Murrinh-Patha Complex Verbs: Syntactic Theory and Computational Implementation

Dissertation submitted for the degree of Doctor of Philosophy

presented by Melanie Seiss

at the Universität Konstanz

Faculty of Humanities
Department of Linguistics

Date of the oral examination: November 25, 2013
First referee: Prof. Dr. Miriam Butt
Second referee: Assoc. Prof. Dr. Rachel Nordlinger
Contents

1 Introduction
  1.1 A brief introduction to Murrinh-Patha .......... 3
  1.2 Formal background ................................ 11
  1.3 Organization of the thesis ...................... 16

2 Complex predicates crosslinguistically ......... 19
  2.1 An overview over different complex predicate constructions . 20
  2.2 Light verbs, serial verbs and auxiliaries ............ 24
    2.2.1 The problem of serial verbs ................... 25
    2.2.2 Auxiliaries and their historical development .... 28
    2.2.3 Light verbs vs. auxiliaries ................... 31
  2.3 Analyses of complex predicates .................... 33
  2.4 Australian coverb constructions .................. 41
    2.4.1 Wagiman complex predicates (Wilson 1999) ...... 47
    2.4.2 Baker & Harvey (2010): Merger versus coindexation . 52
    2.4.3 Jaminjung complex predicates (Schultze-Berndt 2000) 54
    2.4.4 Bardi complex predicates and Bowern’s (2004) analysis 58
    2.4.5 The Ngan’gityemerri system (Reid 2000) .......... 60
    2.4.6 The verbal classification approach (McGregor 2002) . 66
  2.5 Conclusion ......................................... 67

3 Murrinh-Patha complex predicate patterns ....... 69
  3.1 Valency of classifier and lexical stems ............ 72
  3.2 Causatives and anticausatives ..................... 77
  3.3 Intransitive classifier stems in transitive complex predicates . 80
  3.4 Transitive classifier stems in intransitive complex predicates . 86
  3.5 The classifier stem FEET(7): An example of an ambiguous classifier stem .................................. 88
  3.6 Reflexive/reciprocal classifier stems ................ 91
  3.7 Conclusion ......................................... 98
## Contents

4 **An LCS blueprint analysis of complex predicates**  
4.1 Previous approaches  
  4.1.1 Previous LCS based approaches and difficulties in the adaptation to Murrinh-Patha  
  4.1.2 The TCL approach by Caudal et al. (2013a)  
4.2 Hierarchies of selectional restrictions for  
  BASH(14), POKE(19) and SLASH(23)  
  4.2.1 Complex predicates of caused contact  
  4.2.2 Complex predicates of movement  
  4.2.3 Mouth-associated complex predicates  
  4.2.4 Multiple dimensions for classifier stems  
4.3 An LCS blueprint analysis  
  4.3.1 Motivation for using LCSs  
  4.3.2 Blueprints for classifier and lexical stem combinations  
  4.3.3 LCS blueprints and hierarchies of selectional restrictions  
4.4 Conclusion

5 **A computational implementation of the morphology**  
5.1 Challenges posed by the Murrinh-Patha verbal template  
5.2 Implementation of a finite state morphological analyzer  
5.3 Developing a robust morphological analyzer with a stepwise lookup strategy  
5.4 New verbal candidates from the corpus  
  5.4.1 New combinations of classifier and lexical stems  
  5.4.2 New lexical stem candidates  
5.5 Conclusion

6 **An XLE grammar for Murrinh-Patha and applications**  
6.1 An XLE grammar for Murrinh-Patha  
  6.1.1 XLE grammars and the ParGram community  
  6.1.2 The general outline for the morphology-syntax interface  
  6.1.3 The morphology-syntax interface for the complex Murrinh-Patha number system  
  6.1.4 Implementing Murrinh-Patha verbs  
  6.1.5 Implementing free word order in Murrinh-Patha and the interpretation of the dependents  
  6.1.6 Conclusion  
6.2 Applications for learners of Murrinh-Patha  
  6.2.1 Motivation  
  6.2.2 The Murrinh-Patha to English dictionary  
  6.2.3 The English to Murrinh-Patha translation system  
  6.2.4 Further applications  
  6.2.5 Conclusion

7 Conclusion
List of Tables

1.1 Murrinh-Patha verbal template................................. 4
1.2 Paradigm for the classifier stem see(13) .................. 6
1.3 Murrinh-Patha subject number system ..................... 8
1.4 Murrinh-Patha direct object and benefactive markers .... 10

2.1 Ngan’gityemerri classifier stems............................ 61
2.2 Argument structure of Ngan’gityemerri classifier plus lexical stem combinations .................. 62

3.1 Murrinh-Patha classifier stems and their assumed valency . 75
3.2 Correspondences between transitive and RR classifier stems 92

5.1 Murrinh-Patha verbal template................................. 150
5.2 Paradigm for the classifier stem see(13) .................. 151
5.3 Development and test corpus versions of the Bible corpus . 166
5.4 Number of lexical entries for the morphological analyzer . 167
5.5 Output of the morphological analyzer with three strategies for the development corpus word lists .................. 171
5.6 Output of the final morphological analyzer with four lookup strategies for the development corpus word list ........ 174
5.7 Output for the development corpus running text ............ 175
5.8 Output for the test corpus word list ......................... 176
5.9 Output for the test corpus running text .................... 177
5.10 Correspondences between transitive and RR classifier stems 180
5.11 New classifier and lexical stem combinations: RR versions 181
5.12 New classifier and lexical stem combinations: posture/motion 185
5.13 New classifier and lexical stem combinations which occur at least twice .................. 187
5.14 New lexical stem candidates: Spelling alternatives ........ 191
5.15 New lexical stem candidates: Similar lexical items ........ 192
5.16 New lexical stem candidates which occur at least twice .... 194

6.1 Overall architecture of the computational approach ........ 205
6.2 Classifier and lexical stem combinations implemented in the Murrinh-Patha XLE grammar .................. 214
List of Tables
# List of Figures

4.1 Hierarchy of selectional restrictions for complex predicates of caused contact ........................................ 117
4.2 Hierarchy of selectional restrictions for complex predicates of movement ............................................. 120
4.3 Hierarchy of selectional restrictions for complex predicates of “mouth-associated” actions .......................... 123
4.4 An example of the linking between LCS and f-structure ........................................ 135
4.5 Hierarchy of selectional restrictions for complex predicates of caused contact with links to LCS blueprints ........ 136
4.6 Hierarchy of selectional restrictions for complex predicates of movement with links to LCS blueprints ........... 139
4.7 Overall assumed architecture of the lexicon and its links to world knowledge and syntax ............................. 142

5.1 Stepwise lookup strategy for the Murrinh-Patha morphological analyzer for the development corpus ............... 170
5.2 Final stepwise lookup strategy for the Murrinh-Patha morphological analyzer ........................................ 174

6.1 Example output of the English ParGram grammar ........................................ 201
6.2 F-structure as generated by the xle grammar for bamkardu ................................ 212
6.3 Inheritance hierarchy for Murrinh-Patha verb templates ................................ 215
6.4 Output of the Murrinh-Patha electronic dictionary ................................ 224
6.5 The English to Murrinh-Patha translation system ................................ 226
6.6 Output of the English to Murrinh-Patha translation system ................................ 227
6.7 Output of the phonological form checker ................................ 229
6.8 Further applications ................................ 230
Abbreviations

The following abbreviations are used in the glosses:

A    Agentive subject
ACC  Accusative
APPL Applicative
ASP  Unmarked for aspect
BEN  Marker for benefactives, recipients and goals; used for arguments as well as for adjuncts
CAUS Causative
DAT  Dative
DEM  Demonstrative
DM   Discourse marker
DO   Direct object marker
DS   Discourse suffix
DU   Dual
DYN  Dynamic
ERG  Ergative
EXIST Existential
F    Female
FOC  Focus
FUT  Future tense
FUTIRR Future irrealis
GER  Gerund
IBP  Incorporated Body Part
IMP  Imperative
IMPF Imperfective aspect
INCL Inclusive
INF  Infinitive
INSTR Instrument
LOC  Locative
M    Male
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{NC}_{anim}</td>
<td>Noun class marker for non-Aboriginal people and all other animates, including their products</td>
</tr>
<tr>
<td>\textit{NC}_{place}</td>
<td>Noun class marker for time and space</td>
</tr>
<tr>
<td>\textit{NC}_{human}</td>
<td>Noun class marker for Aboriginal people</td>
</tr>
<tr>
<td>\textit{NC}_{fire}</td>
<td>Noun class marker for fire and associated concepts</td>
</tr>
<tr>
<td>\textit{NC}_{res}</td>
<td>Noun class marker for the residue category</td>
</tr>
<tr>
<td>\textit{NC}_{veg}</td>
<td>Noun class marker for flowers, plants and vegetable food</td>
</tr>
<tr>
<td>\textit{NC}_{water}</td>
<td>Noun class marker for fresh water and associated concepts</td>
</tr>
<tr>
<td>\textit{NC}_{weapon}</td>
<td>Noun class marker for offensive weapons</td>
</tr>
<tr>
<td>\textit{NC}_{speech}</td>
<td>Noun class marker for speech and language</td>
</tr>
<tr>
<td>\textit{NFUT}</td>
<td>Non-future tense</td>
</tr>
<tr>
<td>\textit{NOM}</td>
<td>Nominative</td>
</tr>
<tr>
<td>\textit{OBL}</td>
<td>Oblique</td>
</tr>
<tr>
<td>\textit{PAUC}</td>
<td>Paucal</td>
</tr>
<tr>
<td>\textit{PFV}</td>
<td>Perfective aspect</td>
</tr>
<tr>
<td>\textit{PIMPF}</td>
<td>Past Imperfective tense</td>
</tr>
<tr>
<td>\textit{PL}</td>
<td>Plural</td>
</tr>
<tr>
<td>\textit{PPFV}</td>
<td>Past perfective</td>
</tr>
<tr>
<td>\textit{PRS}</td>
<td>Present tense</td>
</tr>
<tr>
<td>\textit{PST}</td>
<td>Past tense</td>
</tr>
<tr>
<td>\textit{PSTIRR}</td>
<td>Past irrealis</td>
</tr>
<tr>
<td>\textit{RDP}</td>
<td>Reduplicated</td>
</tr>
<tr>
<td>\textit{REFL}</td>
<td>Reflexive</td>
</tr>
<tr>
<td>\textit{RR}</td>
<td>Reflexive / Reciprocal</td>
</tr>
<tr>
<td>\textit{S}</td>
<td>Subject (in Ngan'gityemerri, thematic subject)</td>
</tr>
<tr>
<td>\textit{SFOC2}</td>
<td>Emphatic sentence focus</td>
</tr>
<tr>
<td>\textit{SG}</td>
<td>Singular</td>
</tr>
<tr>
<td>\textit{SIBL}</td>
<td>Sibling</td>
</tr>
<tr>
<td>\textit{STAT}</td>
<td>Stative</td>
</tr>
<tr>
<td>\textit{TOP}</td>
<td>Topic</td>
</tr>
</tbody>
</table>
Acknowledgements

First of all, my deepest thanks go to my supervisor Miriam Butt. Before I took a course with her, I was a student set with the idea of becoming a Maths and English high school teacher. Although interested in linguistics before, it was only through her that this interest was expanded. By taking more linguistics courses and by working on her projects, a new perspective opened to me. She has accompanied my development from the start to this dissertation, by discussing ideas and commenting on drafts of presentations, papers and this thesis. Her way of thinking about language and of combining theoretical with computational linguistics influenced my work to a great extent.

This thesis would not have been possible without the support from Rachel Nordlinger either. She introduced me to Murrinh-Patha, was always open for my questions and discussions, interested in the computational approach, tested data for me in the field and gave me the opportunity to spend six wonderful weeks at the University of Melbourne. She also provided very valuable comments on the submitted version of the dissertation from which this published version could profit from. I appreciate all this greatly!

I would also like to thank Joe Blythe who kindly allowed me to use his toolbox dictionary and for discussions about Murrinh-Patha. I owe great thanks to the Murrinh-Patha speakers who taught Rachel, Joe and others their fascinating language so that I could profit from the detailed descriptions of Murrinh-Patha grammar.

My fellow PhD students from the University of Konstanz, especially Tina Bögel, Annette Hautli, Anne Gwinner and Sebastian Sulger were a constant support throughout the years. My office mates during my stay at the University of Melbourne made me feel at home immediately. Special thanks go to Kate Horrack and Aidan Wilson who explored Melbourne with me. Meladel Mistica took care of me during a short research stay at the Australian National University. By the time it took for me to publish this thesis, most of them have submitted or even published their own theses. I wish all of them good luck for their following careers.

Although in a quite different field of linguistics, I learned a lot from Heike Zinsmeister and Stefanie Dipper who employed me in their projects. I am also very grateful to the audiences of numerous LFG conferences, ParGram
meetings and other talks that I have given over the years. Special thanks go to Mark Dras for inviting me to give a talk at Macquarie University Sydney and to Patrick Caudal for discussions about lexical semantics and the joint publication.

I am indebted to the Department of Linguistics of the University of Konstanz for employing me throughout the years and to the Landesgraduiertenstipendium for providing financial support from 2009 to 2012.

My partner, my family and my friends all have been very encouraging and understanding. They provided just the right amount of diversion, from the beginning to the end. Many many thanks to all of you!
Murrinh-Patha is a non-Pama-Nyungan language spoken in and around Wadeye in the Northern Territory of Australia. It is a highly polysynthetic language with a complex verbal structure. The Murrinh-Patha verb morphology is not only complex due to the large number of morphemes that a verb may consist of, but also due to a high degree of syncretism that these morphemes display. Additionally, complex interdependencies exist between the morphemes, which has led Nordlinger (2010c) to claim that the Murrinh-Patha verbal system is a truly templatic system. Thus, while the verbal morphology in itself is already quite complex, the system is further complicated by a complex interface to the syntax and semantic, as no one to one correspondence between the morphological encoding and its syntactic and semantic interpretation exists. Murrinh-Patha morphology and its interface to syntax and semantics thus poses many challenges both for theoretical analysis and computational implementation.

A range of interdependencies in the verbal template are discussed in this thesis, but the main focus is on one special interdependency in detail, namely on the interplay of two morphemes which together contribute the core semantic meaning of the verb. These morphemes are referred to as classifier stem and lexical stem and their combinations are considered morphological complex predicates.

Complex predicates have been discussed for many different languages in a range of diverse frameworks (Chapter 2 provides an overview). My treatment of Murrinh-Patha classifier and lexical stem combinations as complex predicates is mostly influenced by work on complex predicates within Lexical-Functional Grammar (LFG) (Bresnan 1982, 2001) such as Butt (1995), Alsina (1996) and Andrews & Manning (1999) and work on complex predicates in Australian languages such as Wilson (1999), Schultze-Berndt (2000) and Baker & Harvey (2010).

In most accounts of complex predication, the constraints on which elements may combine to form a valid complex predicate are cast within argu-
2 Introduction

ment structure constraints. For example, Ramchand (2008a) describes the constraints on the compatibility of elements forming complex predicates in Bangla in terms of unaccusative, unergative and transitive verbs. However, other, more fine-grained factors may play a role in the selection process in complex predicate formation as well. Butt (1995) shows that semantic features such as volitionality and aspectual concepts play a role in determining the combinatory possibilities in complex predicate formation in Urdu.

This thesis argues that in Murrinh-Patha, semantic concepts which would normally be considered encyclopedic knowledge play a prominent role in the selection process of the combinatory possibilities, irrespective of the number of arguments involved. To model these combinations, this thesis proposes a new way of analyzing complex predicates by combining Jackendoff’s (1990) Lexical Conceptual Structures (LCSs) with a relational approach to lexical semantics. The semantic concepts that restrict combinatory possibilities are modeled with hierarchies of selectional restrictions. These hierarchies are then used to derive the argument structure of the complex predicates in the form of LCS blueprints. Only the combination of both, I claim, can explain both the combination restrictions and the mapping to syntax found in Murrinh-Patha.

The second part of the thesis is concerned with the computational implementation of Murrinh-Patha, with a special focus on the implementation of classifier and lexical stem combinations. The main underlying idea of the theoretic analysis is also reflected in the computational implementation. Looking at linguistic theory and computational implementation for the same data may reveal in which aspects both approaches can profit from each other. Computational implementations profit highly from theoretical insights. This is especially true for rule-based implementations that use a computational version of a theoretical framework as a development platform. One example of such a tight connection is the grammar framework LFG and the grammar development platform XLE (Crouch et al. 2015).

On the other hand, computational implementations can also help gain new theoretical insights. As Bender (2008, 2010) points out, natural language processing can help with hypothesis testing for theoretical analyses. If competing analyses exist, the consequences of one or the other analysis for the whole description of the grammar may not be obvious. A computational implementation of the various analyses may then help to make interactions of the chosen analysis with other parts of the grammar visible.

This thesis presents a rule-based modeling of the Murrinh-Patha morphology and syntax, with a special emphasis on the complex verbal structure. It presents an implementation of the Murrinh-Patha morphology implemented with the Xerox finite-state technology tools XFST and LEXC (Beesley & Karttunen 2003). This morphology is also used in a computational implementation of some parts of the Murrinh-Patha syntax, using the grammar development platform XLE (Crouch et al. 2015).
These computational implementations profited highly from detailed theoretical linguistic descriptions of the Murrinh-Patha morphology and syntax. That is, such an implementation would not be possible without prior thorough descriptions of the language, such as can be found, among others, in Street (1989), Blythe (2009a), Nordlinger (2010a,c), Nordlinger & Caudal (2012) and Walsh (1987, 1997).

The implementations have been used for hypothesis generation and testing as well as testing competing analyses. The hypothesis generation and testing is described detailedly for the development of a Murrinh-Patha morphological analyzer which is used in a corpus study of a small Murrinh-Patha Bible corpus. Because of the complexities of the Murrinh-Patha verbal template, a corpus cannot be used easily without prior morphological processing. With a morphologically analyzed corpus, however, existing hypotheses concerning the Murrinh-Patha morphology can be checked in the corpus and new hypotheses can be generated from the corpus. These new hypotheses can then be tested in future fieldwork. The incorporation of the morphological analyzer into the xle grammar was used to test competing analyses of the Murrinh-Patha complex number system and to evaluate which conclusions can be drawn for the syntax morphology interface. A computational treatment thus makes a highly systematic analysis of the data possible or at least considerably easier.

An implementation can serve further purposes, too. For example, it can be used to build various applications for users which do not require a detailed knowledge of the language or a knowledge about the computational platforms used in the implementations. The Murrinh-Patha grammar implemented with xle, for example, is used in a small translation system and electronic dictionary for beginning learners of Murrinh-Patha. Looking at complex linguistic data both from the perspective of theoretical linguistics and with the aim of designing computational implementations thus has many benefits.

The remainder of this chapter gives an introduction to the morphology and syntax of Murrinh-Patha as needed in this thesis (Section 1.1) and discusses the formal background assumed (Section 1.2). Section 1.3 provides an overview over the organization of this thesis.

1.1 A brief introduction to Murrinh-Patha

Murrinh-Patha is spoken by approximately 2500 people in and around Wad-eye (Port Keats), a small community approximately 400 kilometers south of Darwin in the Daly river region. As has been documented by Kelly et al. (2010), the language is, despite its actual small number of speakers, not considered endangered by the Murrinh-Patha speakers themselves. The current number of speakers is considerable more than pre-contact (Nordlinger 2009).
4 Introduction

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class. SubjN/ Obj</td>
<td>RR</td>
<td>IBP</td>
<td>Lex</td>
<td>TNS</td>
<td>Adv/Prt</td>
<td>SubjN/ Obj</td>
<td>Adv/Prt</td>
<td></td>
</tr>
<tr>
<td>APPL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class: classifier stem, marked for tense, aspect & subject number
SubjN: subject number markers for dual & paucal subject
Obj/ObjN: object markers / object number markers
RR: reflexive / reciprocal marker
IBP: incorporated body part
Lex: lexical stem
TNS: tense marker
Adv/Prt: Adverbial / Particle

Table 1.1: Murrinh-Patha verbal template (adapted from Nordlinger (2010c))

The language is used in every day life and is also acquired by children. It was traditionally considered a language isolate (Tryon 1974), but Green (2003) showed that it can be considered to form a subgroup, called the Southern Daly subgroup, with Ngan’gitymerri (e.g. Reid 1990).

Murrinh-Patha is a highly polysynthetic, head-marking language. The main focus of this work is on verbs, so only some brief remarks on nouns and other parts of speech are provided. As Blythe (2009a) states, case marking on nouns is rare in Murrinh-Patha. More often, discourse markers such as topic or focus markers can be found. A noun phrase may be headed by pronouns or nouns. If a noun is absent, adjectives and demonstratives can function as heads to noun phrases, too (Blythe 2009a). There are ten different noun classes in Murrinh-Patha, denoting concepts such as Aboriginal people, non-Aboriginal people and other animates, a water class, a vegetable class etc. (Walsh 1997).

The Murrinh-Patha verbal morphology, in contrast to the noun morphology, and typically for head-marking languages, is rather complex. A brief overview over the complexities of the verb is given here in the introductory chapter. More detailed descriptions of the data are provided in the remaining chapters of the thesis.

The Murrinh-Patha verb can consist of up to 9 different morphemes which are organized in a verbal template, as is provided in Table 1.1. The

---

1In fact, some verbs can even have an additional RR marker between slot 1 and 2 (c.f. Blythe 2009a). As this is very rarely the case, the template in Table 1.1 ignores this additional template slot.
core semantic meaning of the verb is determined by two lexemes, the classifier stem in slot number 1 and the lexical stem in slot number 5. Examples of verbs consisting solely of these two morphemes are given in (1.1) and (1.2). In (1.1), the classifier stem *see*(13) combines with the lexical stem *ngardu* ‘see’. In this case, it is quite difficult to factor apart what meaning part is contributed by the classifier stem and what meaning part is contributed by the lexical stem. In contrast, the minimal pair in (1.2) allows for deducing the meaning part that is contributed by the lexical stem *rirda* ‘push’ and the meaning parts that are contributed by the classifier stems *hands*(8) and *feet*(7). How the argument structure and meaning of the complex predicate are built by the classifier and lexical stem will be discussed in Chapter 3 and 4.

(1.1) *bamkardu*

bam-ngardu
3SGS.SEE(13).NFUT-see
‘He/She saw him/her.’ (Nordlinger 2011:711)

(1.2) a. *marntirda*

mam-rirda
3SGS.HANDS(8).FUT-push
‘He pushed him (with his hands).’ (Nordlinger 2008)

b. *nungarntirda*

nungam-rirda
3SGS.FEET(7).FUT-push
‘He kicked him.’ (Nordlinger 2008)

The classifier stems, e.g., *see*(13) in example (1.1), are inflected for subject person and number and for tense. Most of the classifier stems cannot form a predicate on their own, they have to combine with lexical stems to form a predicate. As it is difficult to determine the meaning of these classifier stems for this reason, the classifier stems have traditionally been glossed with a number. For some of the classifier stems a meaning could be established by now, such as *see* for the classifier stem 13. For ease of reference and compatibility with other publications on Murrinh-Patha, the number is retained in the gloss (c.f. Nordlinger 2015).

There are most probably 38 different classifier stems in Murrinh-Patha, and each of them has an extensive verbal paradigm, inflecting for four different number and five different tense categories in portmanteau forms (c.f. Blythe et al. 2007). The paradigm for the classifier stem *see*(13) is provided in Table 1.2 as an example.

The lexical stems, e.g., *ngardu* ‘see’ in (1.1), are uninflecting. They also form a considerably larger class than the classifier stems, with numbers in the hundreds. Lexical stems can never occur as predicates on their own, they
always have to combine with a classifier stem (c.f., e.g. Nordlinger 2015). As such, the meaning of the lexical stem can only be determined in combination with the classifier stem. Some lexical stems combine with a whole range of classifiers stems. For these, a meaning can be established independently of the classifier stem. But for the lexical stems which only occur with one classifier stem, the gloss reflects the meaning of the lexical stem with this particular classifier stem. This should be kept in mind when looking at the glosses of the lexical stem.

Apart from classifier and lexical stems, tense markers may be part of the verbal template, too. Most often, the separate tense markers in slot number 6 need to agree with the tempus marking on the classifier stem, e.g., the past imperfective form of the classifier stem requires the tense marker -dha in slot number 6. (1.3) provides an example involving the future form of the classifier stem bash(14) and the future tense marker -nu.

(1.3)  
\textit{ba-warta-nu}  
\textit{1SG.SING.BASH(14).FUT-split.open-FUT}  
‘I’ll split it open (with an axe).’  
(Nordlinger & Caudal 2012:80)

Nordlinger & Caudal (2012) provide a detailed overview over the tense, aspect and modality system in Murrinh-Patha by discussing the various combinatorial possibilities of the classifier stem with the separate tense markers in slot number 6. They also discuss the interaction of the tense marking on the verbal complex with serialized classifier stems which can cliticize onto the

<table>
<thead>
<tr>
<th></th>
<th>nFut</th>
<th>PImpf</th>
<th>Fut</th>
<th>FutIrr</th>
<th>PstIrr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sing</td>
<td>1</td>
<td>bam</td>
<td>be</td>
<td>ba</td>
<td>ba</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>dam</td>
<td>de</td>
<td>da</td>
<td>da</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>bam</td>
<td>be</td>
<td>ba</td>
<td>ba</td>
</tr>
<tr>
<td>Dual</td>
<td>1</td>
<td>ngubamka</td>
<td>ngube</td>
<td>nguba</td>
<td>nguba</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>nubamka</td>
<td>nube</td>
<td>muba</td>
<td>muba</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>pubamka</td>
<td>pube</td>
<td>puba</td>
<td>kuba</td>
</tr>
<tr>
<td>Plur</td>
<td>1</td>
<td>ngubam</td>
<td>ngube</td>
<td>nguba</td>
<td>nguba</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>nubam</td>
<td>nube</td>
<td>muba</td>
<td>muba</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>pubam</td>
<td>pube</td>
<td>puba</td>
<td>kuba</td>
</tr>
</tbody>
</table>

Table 1.2: The paradigm for the classifier stem see(13), taken from Blythe et al. (2007)
already existing classifier and lexical stem combination. This serialized use of the classifier stems is restricted to intransitive classifier stems of posture and motion. In (1.4), the classifier stem STAND(3) attaches to the classifier and lexical stem combination, resulting in a different tense and aspect specification.

(1.4) a. nungarnirda
    nungam-rirda
    3SGS.FEET(7).NFUT-push
    ‘He kicked him.’ (Nordlinger & Caudal 2012:95)

    a. nungarnirda=pirrim
    nungam-rirda=pirrim
    3SGS.FEET(7).NFUT-push=3SGS.STAND(3).NFUT
    ‘He’s kicking him.’ (Nordlinger & Caudal 2012:95)

There also seems to be a correlation between tense and aspect marking as discussed so far and the use of reduplicated lexical stems which seems to be able to contribute aspectral information (Street 1980a). In (1.5), the lexical stem kum and its reduplicated form kumkum are used in combination with the classifier stem go(6). With the reduplicated form of the lexical stem as in (1.5b), a continuous or repetition reading arises.

(1.5) a. ngurrur-kum-nu
    1SGS.GO(6).FUT-swim-FUT
    ‘I’ll swim.’ (Street 1989)

    b. ngurrur-kumkum
    1SGS.GO(6).NFUT-swim(RDP)
    ‘I’m continually swimming.’ (Street 1989)

In some cases, the reduplicated form also contributes a sense of a plural object as in (1.6) (Street 1980a). Although an interesting topic, this thesis will not be concerned with the semantics of reduplications. However, to be able to understand the glosses of reduplicated lexical stems in the remainder of the thesis, a few words on their form are required.

(1.6) a. ma-dharl-nu
    1SGS.HANDS(8).FUT-open-FUT
    ‘I will open it.’ (Street 1989)

    b. mantharlurl
    mam-dharlurl
    3SGS.HANDS(8).NFUT-open(RDP)
    ‘I opened them (many objects).’ (Street 1989)
As can be seen in (1.5) and (1.6), reduplication may involve a range of different forms. (1.5) presents an example of a full reduplication while in (1.6), the lexical stem is only partially reduplicated and additionally, the vowel is changed. A further quite common pattern seems to be to add the final coda of the lexical stem to the penultimate syllable of the lexical stem, such as in the lexical stem *dharrmpurl* ‘mark a tree’ with its reduplicated form *dharrmurlpurl*. In some cases, assimilation may then arise, such as in the stem *dhadum* ‘sink’ whose reduplicated version is *dhamnum*.

Although Street (1980a) discusses some generalizations on the formation of reduplication, the reduplication of lexical stems seems to be lexicalized to a high degree, so that Street (1989) actually lists the reduplicated forms together with the lexical entries of the lexical stems. Street (1989) also includes as reduplicated lexical stems instances of lexical stems which substitute for another lexical stem to convey the meaning of repetitive action or of plurality, according to Street (1980a). Such a pair is, for example, *winhikat, winhiyerrarr* ‘make a mistake’. In the remainder of the thesis, the glosses indicate whether the lexical stems are listed as reduplicated forms of another lexical stem by Street (1989).

Turning back to the verbal template as such, subject number markers can be found in slots number 2 and 8 which combine with the information on the classifier stem to form a complex number system, grammaticalizing gender as well as siblinghood as part of the contrast. The system can be schematized as in Table 1.3. The classifier stem has a three-way number contrast (singular, dual or plural). However, the dual markers *ngintha(f)/nintha(m)* can attach to the singular classifier stem to denote a dual non-sibling subject. Similarly, the paucal markers *ngime(f)/neme(m)* can attach to the dual classifier stem to form a paucal non-sibling subject. Constructed examples, taken from Nordlinger (2011), are provided in (1.7).

(1.7) a. *bam-ngintha-ngkardu*

3SG.SEE(13).NFUT-DU.F-see

‘They two (female non-siblings) saw him/her.’
b.  *pubamka-ngkardu*
   
   3DU.S.see(13).NFUT-see
   
   ‘They two (siblings) saw him/her.’

c.  *pubamka-ngkardu-ngime*
   
   3DU.S.see(13).NFUT-see-PAUC.F
   
   ‘They paucal (female non-siblings) saw him/her.’

d.  *pubam-ngkardu*
   
   3PLS.See(13).NFUT-see
   
   ‘They (paucal siblings/plural) saw him/her.’

(Nordlinger 2011:711)

A more detailed discussion of the data, along with an LFG implementation, can be found in Nordlinger (2010a, 2012b). Section 6.1.2 discusses these analyses in the light of a complex morphology syntax interface, proposes a computational implementation and lays out how a computational implementation may help gain insights into the complexity of the system.

As is typical for head-marking languages, other sentence participants are marked on the verb as well. In Murrinh-Patha, direct objects as well as ‘benefactives’, i.e., arguments or adjuncts encoding a benefactive, recipient or goal, are marked on the verbal template (Nordlinger 2011). The direct object and benefactive markers display a five-way number contrast as well as gender and siblinghood contrasts, too. Table 1.4 provides an overview over the different forms of direct object and benefactive markers.

As can be seen in Table 1.4, the markers for dual and paucal non-sibling objects and benefactives are encoded with a combination of two morphemes. In this case, the first morpheme occupies slot number 2 while the dual and paucal number markers -ngintha(f)/-nintha(m) and -ngime(f)/-neme(m) occupy slot 8. (1.8) provides some examples. In (1.8a), the direct object marker - nhi is used while in (1.8b), the benefactive marker -mpa is used. (1.8c) is an example involving a paucal female direct object marker which consists of the marker -nku, encoding third person direct object, and the paucal female number marker -ngime.

(1.8)  

a.  *nga-nhi-berti-nu*  
   
   3SGS.HANDS(8).FUT-2SG.DO-take-FUT  
   
   ‘I’ll take you home.’  
   
   (Nordlinger 2011:710)

b.  *nga-mpa-berti-nu*  
   
   3SGS.HANDS(8).FUT-2SG.BEN-take-FUT  
   
   ‘I’ll take him home for you.’  
   
   (Nordlinger 2011:710)

c.  *dirra-nku-wintharrarr-dha-ngime*  
   
   3SGS.28.PIMPF-3DO-seek-PIMPF-PAUC.F  
   
   ‘He was looking for the few women.’  
   
   (Nordlinger 2010c)
<table>
<thead>
<tr>
<th></th>
<th>Direct object</th>
<th>Benefactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-ngi</td>
<td>-nga</td>
</tr>
<tr>
<td>2</td>
<td>-nhi</td>
<td>-mpa</td>
</tr>
<tr>
<td>3f</td>
<td>-</td>
<td>-nge</td>
</tr>
<tr>
<td>3m</td>
<td>-</td>
<td>-na</td>
</tr>
<tr>
<td>1 Incl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>du</td>
<td>-nhi</td>
<td>-nhe</td>
</tr>
<tr>
<td>pauc f</td>
<td>-nhi -ngime</td>
<td>-nhe -ngime</td>
</tr>
<tr>
<td>pauc m</td>
<td>-nhi -neme</td>
<td>-nhe -neme</td>
</tr>
<tr>
<td>pl</td>
<td>-nhi</td>
<td>-nhe</td>
</tr>
<tr>
<td>Dual nonsibl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1f</td>
<td>-nganku -ngintha</td>
<td>-ngarru -ngintha</td>
</tr>
<tr>
<td>1m</td>
<td>-nganku -nintha</td>
<td>-ngarru -nintha</td>
</tr>
<tr>
<td>2f</td>
<td>-nanku -ngintha</td>
<td>-narru -ngintha</td>
</tr>
<tr>
<td>2m</td>
<td>-nanku -nintha</td>
<td>-narru -nintha</td>
</tr>
<tr>
<td>3f</td>
<td>-nk -ngintha</td>
<td>-rru -ngintha</td>
</tr>
<tr>
<td>3m</td>
<td>-punku -ngintha*</td>
<td>-pirru -ngintha*</td>
</tr>
<tr>
<td>Pauc nonsibl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1f</td>
<td>-nganku -ngime</td>
<td>-ngarru -ngime</td>
</tr>
<tr>
<td>1m</td>
<td>-nganku -neme</td>
<td>-ngarru -neme</td>
</tr>
<tr>
<td>2f</td>
<td>-nanku -ngime</td>
<td>-narru -ngime</td>
</tr>
<tr>
<td>2m</td>
<td>-nanku -neme</td>
<td>-narru -neme</td>
</tr>
<tr>
<td>3f</td>
<td>-nk -ngime</td>
<td>-rru -ngime</td>
</tr>
<tr>
<td>3m</td>
<td>-punku -ngime*</td>
<td>-pirru -ngime*</td>
</tr>
<tr>
<td>Pauc sibl and Plural</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-ngan</td>
<td>-ngarra</td>
</tr>
<tr>
<td>2</td>
<td>-nan</td>
<td>-narra</td>
</tr>
<tr>
<td>3</td>
<td>-n</td>
<td>-rru</td>
</tr>
<tr>
<td></td>
<td>-pun*</td>
<td>-pirra*</td>
</tr>
</tbody>
</table>

Table 1.4: Murrinh-Patha direct object and benefactive markers, adapted from Blythe et al. (2007). Forms marked with * are used only with non-future verbs.
This means that the subject number markers as well as the direct object and benefactive markers compete for the same slots, as has been laid out in Blythe (2009a) and Nordlinger (2010c). Chapter 5 presents these interdependencies in more detail and discusses how these interdependencies make a computational implementation of the morphology challenging.

The verbal template further comprises a slot for a reflexive/reciprocal marker (for details see Nordlinger 2008), a slot for incorporated body parts (Walsh 1995) which also hosts the applicative marker -ma (Nordlinger 2009) as well as two slots for particles and adverbs. In addition to the complexity based on the number of different morphemes and the interdependencies, morphophonemic processes such as seen in (1.1), (1.2) or (1.4) make the morphology of Murrinh-Patha verbs complex.


1.2 Formal background

Complex predicates have been discussed in a range of different theoretic frameworks as they challenge fundamental views on the division of work between the lexicon, syntax and semantics. This thesis is most prominently influenced by the work on complex predicates cast within Lexical-Functional Grammar. Approaches to complex predicates within LFG are, for example, Butt (1995), Alsina (1996, 1997), Wilson (1999) or Andrews & Manning (1999).

LFG has also proven very useful in the analysis of Australian languages generally, as was shown for example by Simpson (1991), Austin & Bresnan (1996) or Nordlinger (1998). LFG is therefore here adopted as a theoretical background, although the framework itself is not very prominent in most parts of the thesis. However, many assumptions made in this work are influenced by the framework. This section therefore gives a brief overview over the formalism and the assumptions the formalism makes about grammar.

As Asudeh & Toivonen (2009) lay out, the core ideas of the formalism of LFG were developed in the late 1970s by Joan Bresnan and Ron Kaplan who wanted to design a linguistic theory that was both psychologically plausible and mathematically calculable. The formalism as it was developed at that time was published in Bresnan (1982).

LFG is a generative theory of grammar with a modular architecture, i.e., it assumes various levels of data structures which, in contrast to most other grammar theories, can make use of very different principles and represen-
LFG is most prominently seen as a theory of syntax, for which it proposes two distinct structures for analysis, the f(unctional)-structure and the c(ontinuum)-structure. These two central structures were proposed in Bresnan (1982) already and have not been changed considerably since then.

In addition to these two structures, other parallel structures have been proposed in more recent years which make LFG a theory of grammar instead of pure syntax. Among these structures are a(rgument)-structure, s(emantic)-structure (e.g., Dalrymple 2001), p(honological)-structure (e.g., Dalrymple & Mycock 2011, Bögel et al. 2010, Bögel 2015) and i(nformation)-structure (e.g., Dalrymple & Nikolaeva 2011). While the theory is stable in the sense that the core ideas of the formalism developed in the 1970s are still the fundamental assumptions of the theory today, the theory is also flexible enough to incorporate these newer levels. In the following, I briefly present the levels that are important for this thesis.

The f- and c-structures are the structures assumed by LFG for syntactic analysis. C-structures are represented as tree structures modeling linear word order, constituency etc. In contrast to transformation-based approaches, LFG assumes that these tree structures are base generated, i.e., that they are pure representations of the surface structure of sentences. LFG usually assumes an X'-scheme but also allows for an exocentric category S, i.e., a phrase lacking a head. The terminal nodes of the c-structure have to be filled with morphologically complete words, which is modeled by the Principle of Lexical Integrity.

As the c-structure represents the surface order of strings, it might look very differently across languages. The c-structure of the Murrinh-Patha sentence in (1.9) is illustrated in (1.10a). As Murrinh-Patha is a highly polysynthetic language in which the subject as well as the object can be expressed by verbal inflection without the need of expressing the arguments overtly, the c-structure is very simple. In contrast, the English translation of the Murrinh-Patha example with the overtly expressed noun phrases has the c-structure in (1.10b).

(1.9) bamkardu
bam-ngkardu
3SGS.SEE(13).NFUT-see
‘He saw her.’
While the c-structure may be guided by very different principles based on the specific properties of the languages, the f-structure is assumed to represent invariant, crosslinguistically valid principles. It is called functional structure because it represents grammatical functions such as the subject and the object of a sentence. The f-structure is modeled as an attribute-value matrix which needs to adhere to the Uniqueness or Consistency Principle, which specifies that each attribute has exactly one value.

(1.11) provides a simplified f-structure for the English c-structures in (1.10). The analysis for the Murrinh-Patha f-structure is not as straightforward and will be discussed in Section 6.1.4. However, it is still quite similar to the English f-structure. The fact that f-structures provide a deep syntactic analysis which is the same or very similar across languages is exploited in computational approaches for machine translation. This is further explained in Section 6.1.1.

As can be seen in (1.11), the verb ‘see’ requires a subject and an object. How many arguments a verb requires is specified by the lexical entry of the verb. The well-formedness Principles of Completeness and Coherence then ensure that a sentence only receives a grammatical analysis if all the arguments that a verb requires (and not more) are present. These principles thus identify sentences as in (1.12) as ungrammatical.

(1.12) a. *He saw.

b. *He saw her the girl.

The c-structure and f-structure of a sentence are identified through a range of annotations on the c-structure. The underlying idea is that each node of the c-structure corresponds to a specific part of the f-structure, e.g., the
first NP of the English sentence in (1.10b) corresponds to the subject of the sentence.

How these annotations are arrived at may depend from language to language. In configurational languages like English, a specific node may be associated with the relevant information, e.g., the sister of VP may be identified as contributing the subject while the sister of V may contribute the object. This is illustrated in the annotated c-structure in (1.13).

The annotation ↑↓ is used for heads and passes the information of the annotated node up to the mother node. In this way, then, the information of the verb, for example see, ends up as information of the whole sentence S. The notation ↑subj↓ ensures that the information that is contained in the annotated node is interpreted as the subject of the sentence S.

(1.13)

While the information of the grammatical functions is encoded directly in the c-structure positions for configurational languages, for non-configurational languages this information might come from different sources. For example, case marking might directly contribute to the interpretation of the grammatical functions, as was argued for in Nordlinger (1998) and Butt (2006).

The division of labour between the f- and c-structure also allows an elegant treatment of discontinuous constituents such as found in many Australian languages. As Austin & Bresnan (1996) and Nordlinger (1998) lay out, discontinuous constituents may occupy different nodes in the c-structure but may be mapped onto the same grammatical function in the f-structure.

In head-marking languages which allow both the subject and the object to be dropped such as in Murrinh-Patha, it is assumed that the inflection on the verb may provide information about the subject and object involved. In this case, the lexical entry of the verb specifies the information for the subject and the object. (1.14) provides a simplified lexical entry for bamkardu.
This illustrates the idea that ‘morphology competes with syntax’, i.e., that grammatical relations can be encoded by morphological marking or by syntactic principles such as word order (Nordlinger & Bresnan 2011).

(1.14) bāmkardu V (↑ PRED) = ‘see<(SUBJ)(OBJ)>’
(↑ TENSE) = PAST
(↑ SUBJ) = ‘pro’
(↑ OBJ) = ‘pro’

Such complex lexical entries of course pose the question of how these lexical entries are arrived at, i.e., what morphological operations are assumed and how the interface between morphology and syntax is modeled in LFG.

A range of different treatments for morphology and the morphology-syntax interface have been proposed in LFG, which cannot be treated here in detail (but see e.g., Sadler & Spencer 2004, Sadler & Nordlinger 2004, 2006). Chapter 6, however, presents the morphology-syntax interface that is assumed in the computational implementation of Murrinh-Patha and discusses theoretical implications.

LFG also assumes an argument structure that models the interplay between lexical semantics and syntactic structure. Most approaches to argument structure in LFG assume a list of thematic roles to model the mapping between lexical semantics and syntactic structure. Each predicate specifies its argument structure either in terms of how many arguments the predicate takes, such as in (1.15a) or even in terms of what thematic roles are required for the predicate, as in (1.15b).

(1.15) a. ‘see (arg1, arg2)’
   b. ‘see (experiencer, theme)’

In approaches that assume an argument structure such as (1.15a), the thematic information such as the experiencer and theme, or even more concrete roles such as the ‘seer’ and the ‘seen’ are assumed to be part of the lexical semantics instead of part of argument structure (e.g., Mohanan 1997, Bresnan 2001). They model the mapping from lexical semantics to argument structure with the help of a thematic role hierarchy, shown in (1.16).

(1.16) **Thematic role hierarchy**

\[
\text{agent} > \text{beneficiary} > \text{experiencer/goal} > \text{instrument} > \text{patient/theme} > \text{locative}
\]

(Bresnan 2001:307)

Bresnan (2001) explains that the most prominent role in lexical semantics, e.g., the experiencer for see, is the arg1 in (1.15a) while the theme role of see corresponds to arg2. This additional mapping step is often left out and the argument structure is assumed to include the thematic roles directly, such as in (1.15b).
Some research has also suggested different approaches to argument structure. Alsina (1996, 1997) proposes using Dowty’s (1991) Proto-Roles while Butt (1995) uses Lexical Conceptual Structures as presented by Jackendoff (1990) as argument structure. The different approaches to argument structure and their application to complex predicates in different languages are discussed in more detail in Chapter 2.

Along with these different ways of representing argument structure come different approaches of how the argument structure should be mapped onto the syntactic structure, i.e., onto grammatical relations. However, most approaches share the basic assumptions of the standard Lexical Mapping Theory (Bresnan & Zaenen 1990) which assumes binary-valued features [± r] (for restricted) and [± o] (for objective). Grammatical functions are assumed to have the specifications as in (1.17).

\[(1.17)\]

<table>
<thead>
<tr>
<th></th>
<th>-r</th>
<th>+r</th>
</tr>
</thead>
<tbody>
<tr>
<td>-o</td>
<td>SUBJ</td>
<td>OBL_o</td>
</tr>
<tr>
<td>+o</td>
<td>OBJ</td>
<td>OBJ_o</td>
</tr>
</tbody>
</table>

The thematic roles may also receive values for [± r] and [± o] based on different mechanisms so that linking can take place through these features. In this standardly assumed mapping, the mapping from argument structure to grammatical relations happens in the lexicon, and each thematic role must be mapped to exactly one grammatical function. As will be seen in Chapter 2, these assumptions need to be revised when working on the argument structure of complex predicates.

As can thus be seen, the different levels assumed in the LFG framework make use of very different mechanisms for linguistic analysis. The framework itself does not pose strict requirements on the mechanisms that are used in the different levels and assumes that different mechanisms will indeed be required to handle data flexibly. This is one of the strengths of the LFG formalism, as will be discussed in Chapter 6 for the modeling of the morphology and syntax of Murrinh-Patha with distinct formalisms and a complex interface between those.

As was said above, work within LFG is also concerned with further levels of grammatical analysis and their interfaces. However, this thesis will be mostly concerned with the question of morphology, syntax, lexical semantics and argument structure so that the other levels will not be discussed here. For more information on LFG, the reader is referred to Bresnan (1982, 2001), Dalrymple (2001) and Asudeh & Toivonen (2009).

### 1.3 Organization of the thesis

The thesis is structured as follows. Chapter 2 provides an overview over complex predicate constructions crosslinguistically. It lays out what con-
Instructions are considered complex predicates in this work and how they can be distinguished from other multi-verb structures such as serial verbs or auxiliary constructions. It then proceeds to discuss formal analyses of complex predicate constructions in different frameworks.

Special emphasis is given in this chapter to Australian complex predicates formed with an inflecting element such as the Murrinh-Patha classifier stem and an uninflecting element such as the lexical stem, which is often also called the coverb. The chapter discusses the main properties of these constructions and how they differ across various Australian languages. Finally, a range of different analyses proposed for Australian coverb constructions, among them Wilson’s (1999) analysis of Wagiman complex predicates, Schultze-Berndt’s (2000) analysis for Jaminjung, Bowern’s (2004) analysis for Bardi as well as Reid’s (2000) analysis for Ngan’gityemerri are discussed.

Chapter 3 is concerned with the various patterns of complex predication found in Murrinh-Patha. It analyzes the patterns from an argument structure perspective. Most approaches to Australian coverb constructions, or even most approaches to complex predicates crosslinguistically, try to derive the argument structure of the complex predicate as a whole from the argument structure of the elements that form the complex predicate. For this reason, the argument structure is usually considered the first excluding factor in complex predicate formation while other factors such as semantic concepts might then further restrict the combinatory possibilities. This chapter argues that in Murrinh-Patha, the argument structure does not restrict the combinatory possibilities. Instead, specific semantic concepts are responsible for the combinatory possibilities and the argument structures can be derived from these.

Chapter 4 presents the semantic analysis of Murrinh-Patha complex predicates. It first shows that previous approaches to complex predicates making use of LCSs cannot be applied easily to the Murrinh-Patha data, as Murrinh-Patha complex predicates display a too large variety to model these possibilities with a restricted set of LCS rules. It then discusses the classifier stems bash(14), poke(19) and slash(23) as a case study into the finer-grained semantic concepts which are responsible for combinatory possibilities in Murrinh-Patha. The chapter proposes a hierarchy of selectional restrictions that can then be used to derive the argument structure of Murrinh-Patha complex predicates. This is modeled with LCS blueprints that define the behavior of whole classes of classifier and lexical stems in Murrinh-Patha. In contrast to Butt (1995) or Wilson (1999), among others, the LCS are not used to model the combinatory possibilities. However, the chapter argues that LCSs are useful in modeling the argument structure once the combinatory possibilities have been established on semantic grounds.

Chapter 5 is concerned with the computational implementation of the morphological analyzer. For this purpose, the challenges that the Murrinh-Patha verbal template poses for a computational implementation of the
morphology are summarized. It is argued that Xfst (Beesley & Karttunen 2003) offers the tools for an implementation of the morphological analyzer and a stepwise lookup strategy is proposed to ensure both effective and reliable analysis. Finally, the chapter evaluates the output of the morphological analyzer and discusses how a morphologically processed corpus can be combined with other methods to gain new insights into the vocabulary and structure of a language.

In Chapter 6, a range of applications for beginning learners of Murrinh-Patha are presented. These applications use the morphological analyzer presented in Chapter 5 and a computational grammar of Murrinh-Patha, built with the grammar development platform XLE (Crouch et al. 2015). This grammar is presented in the first part of the chapter, with a brief description of the XLE grammar development platform, a discussion of the morphology-syntax interface assumed in the implementation as well as issues in the computational analysis of Murrinh-Patha word order and the implementation of the classifier and lexical stem combinations. The second part of the chapter then summarizes the applications for learners of Murrinh-Patha, i.e., the electronic dictionary, a translation system as well as further applications. Chapter 7 concludes the thesis.
Chapter 2

Complex predicates crosslinguistically

This chapter is concerned with complex predicates crosslinguistically, with a special focus on complex predicates in Australian languages. As Butt (2010) and Baker & Harvey (2010), among others, point out, the term “complex predicate” is used for a wide range of different constructions, and crosslinguistically similar constructions are not necessarily always identified as such by subsuming them under one and the same label, e.g., complex predicates. This chapter thus serves the purpose to make clear what constructions are treated as complex predicates in this work.

Butt (1995) defines a complex predicate in terms of Lexical-Functional Grammar as follows:

(2.1) Definition of a Complex Predicate (Butt 1995)

- The argument structure is complex (two or more semantic heads contribute arguments).
- The grammatical functional structure (f-structure) is that of a simple predicate. It is flat: there is only a single predicate (a nuclear PRED) and a single subject.
- The phrase structure (c-structure) may be either simple or complex. It does not necessarily determine the status of a complex predicate.

Similar definitions can also be found in Mohanan (1994, 1997) or Alsina et al. (1997). Different construction types fall into the scope of this definition of complex predicates. For example, the definition both accounts for verb plus verb as well as for verb plus noun complex predicates. What is referred to as a complex predicate construction can thus look very differently from language to language as well as within a language. Section 2.1 therefore
Section 2.2 discusses how complex predicate constructions can be distinguished from other multi-verb constructions, such as auxiliary verb constructions and serial verbs. Analyses of complex predicates proposed in various linguistic frameworks are presented in Section 2.3. Finally, Section 2.4 is concerned with Australian complex predicates in more detail. After setting out the basic properties that Australian complex predicates share as well as highlighting the main dimensions in which they may differ, a range of analyses of Australian complex predicates are discussed in more detail. Section 2.5 concludes the chapter.

2.1 An overview over different complex predicate constructions

Based on the definition in (2.1), a complex predicate is a construction in which two or more grammatical elements function together to contribute information that is normally associated with a single head. Many different constructions may fall under the label complex predicate based on this definition. Among the most common complex predicates are verb plus verb complex predicates as well as verb plus noun complex predicates. However, complex predicates can also be formed with a verb in combination with an adjective, preposition or a special part-of-speech that can only be used in combination with another verb, such as e.g., coverbs (Amberber et al. 2007).

Verbs that may combine with another part of speech to form complex predicates are usually referred to as light verbs, a term that was coined by Jespersen (1954) for constructions such as in (2.2), taken from Butt (2010).

(2.2) have a rest, a read, a cry, a think
take a sneak, a drive, a walk, a plunge
give a sigh, a shout, a shiver, a pull, a ring

Such English constructions have been studied by Jackendoff (1974) and Cattell (1984), among others. Another early and well-studied example of verb plus noun complex predicates is the Japanese construction involving the verb suru ‘do’. Grimshaw & Mester (1988) point out that the object of suru ‘do’ determines the theta-marking on the other arguments. For example, in (2.3a), the direct object of suru is hanashi ‘talk’, and this object \( \theta \)-marks the argument Mary as a goal or beneficial. In contrast, the direct object shuppatsu ‘departure’ in (2.3b) requires the argument Tookyoo ‘Tokyo’ to be \( \theta \)-marked as source. This is, in these cases the verb suru and its direct object function together to form the predicate of a clause, i.e., they form a complex predicate together.
Complex predicates crosslinguistically

(2.3)  
a. John-wa Mary-ni hanashi-o shita  
   John-TOP Mary-to talk-ACC suru  
   ‘John talked to Mary.’

a. John-wa Tookyoo-kara shuppatsu-o shita  
   John-TOP Tokyo-from departure-ACC suru  
   ‘John departed from Tokyo.’

(Japanese, Grimshaw & Mester 1988:207)

The phenomenon of verb plus verb complex predicates was first discussed by Aissen & Perlmutter (1983) in their treatment of cases of Spanish clause union. They investigated Spanish causatives and similar constructions in which they could show that such sentences are monoclausal in Spanish, despite the fact that they involve two verbs.

(2.4) provides an example for a Spanish causative construction. The clitic object la ‘her’ seems to occur in the matrix clause headed by hice ‘made’. By looking at various tests such as constraints on clitic climbing, reflexive passivization and object raising, Aissen & Perlmutter (1983) show that this construction does not in fact involve clitic climbing, i.e., the clitic does not ‘climb up’ into the higher clause, but that the phenomenon instead is one of clause union, i.e., that there is only one clause for which both verbs form the predicate together.

(2.4)  
la hice correr  
her.ACC I-made run  
‘I made her run.’  
(Spanish, Aissen & Perlmutter 1983:384)

The phenomenon has been discussed for other Romance languages as well. (2.5) is an example from French taken from Rosen (1990). Work on Romance causatives building on these insights includes, among others, Zubizarretta (1985), Guasti (1996), Cinque (1998), Andrews & Manning (1999) and Alsina (1996).

(2.5)  
a. Jean a fait partir Marie.  
   Jean has made go Marie  
   ‘Jean made Marie go.’  
   (French, Rosen 1990:37)

b. Jean l’a fait partir.  
   Jean her has made go  
   ‘Jean made her go.’  
   (French, Rosen 1990:23)

Complex predicates in Hindi (e.g. Hook 1974, Mohanan 1994, 1997) and Urdu (e.g. Butt 1995) have also been extensively studied. Butt (1995) is mainly concerned with verb plus verb complex predicates such as the permissive ((2.6a)) and aspectual complex predicates ((2.6b)) in Urdu.
(2.6) a. \textit{anjum=ne} \quad \textit{saddak=ko} \quad \textit{jaa-ne} \quad \textit{di-yaa}  \\
\text{Anjum.F=ERG} \quad \text{Saddaf.F=DAT} \quad \text{go-INF.OBL} \quad \text{give-PFV.M.SG}  \\
‘Anjum let Saddaf go.’  \\
\text{(Urdu, Butt 1995:35)}  \\

b. \textit{anjum=ne} \quad \textit{xat} \quad \textit{li-k} \quad \textit{li-yaa}  \\
\text{Anjum.F=ERG} \quad \text{letter.M=NOM} \quad \text{write} \quad \text{take-PFV.M.SG}  \\
‘Anjum wrote a letter (completely).’  \\
\text{(Urdu, Butt 1995:90)}

While the permissive in (2.6a) is rather similar to the instances of causatives discussed above, the aspectual complex predicate in (2.6b) poses new questions to the analysis of complex predicates, namely how such constructions can be distinguished from auxiliary constructions. Butt (1995) shows that these aspectual complex predicates in fact differ from auxiliary constructions in Urdu as the light verb has an influence on the argument structure and case marking. The following Section 2.2 discusses the difference between light verbs and auxiliaries in more detail.


Noun plus verb complex predicates have also been studied for Persian. As Folli et al. (2005) lay out, complex predicates are very common in Persian, and the corresponding simple verbs are very often restricted to stylistically high and written language. In Persian complex predicate formation, a range of different light verbs can be used, such as \textit{kardan} ‘do’, \textit{shodan} ‘become’ etc. (Folli et al. 2005). (2.7a,b) shows the verb \textit{zad} ‘hit’ used as a main verb in (2.7a) and as a light verb as part of a complex predicate with the noun \textit{harf} ‘word’ in (2.7b). Complex predicates in Persian are also used to incorporate new concepts into the languages, such as \textit{telephone} in (2.7c).

(2.7) a. \textit{Ali} \quad \textit{mard-r} \quad \textit{zad}  \\
\text{Ali} \quad \text{man-R\textsubscript{A}} \quad \text{hit.3SG}  \\
‘Ali hit the man.’  \\
\text{(Persian, Müller 2010:606)}  \\

b. \textit{Ali} \quad \textit{BAb\acute{a}k} \quad \textit{harf} \quad \textit{zad}  \\
\text{Ali with Bab\acute{a}k word hit.3SG}  \\
‘Ali talked with Bab\acute{a}k.’  \\
\text{(Persian, Müller 2010:606)}  \\

c. \textit{(man)} \quad \textit{telefon} \quad \textit{kard-am}  \\
\text{I} \quad \text{telephone} \quad \text{did-1SG}  \\
‘I telephoned.’  \\
\text{(Persian, Müller 2010:605)}
Persian noun plus verb complex predicates, as well as Persian complex predicates involving other parts of speech such as adjectives, adverbs or prepositions, have also been discussed by e.g., Goldberg (1996, 2003), Karimi (1997), Karimi-Doostan (1998, 2005), Megerdoomian (2001), Mohammad & Karimi (1992), Nemati (2010) and Müller (2010).

While the examples discussed so far all involved complex predicates that were formed with two different lexical items in the syntax, morphological complex predicates also exist. Alsina & Joshi (1991) show that the same patterns concerning direct and indirect causation can be found in morphological complex predicates in Chichewa and syntactically formed causatives such as found in French. (2.8) illustrates the two different kinds of causatives found in Chichewa.

(2.8) a. Nṉgu i-na-phik-ɪtsa kadžidzi maṉgu.
  porcupine S-PST-cook-CAUS owl pumpkins
  ‘The porcupine made the owl cook the pumpkins.

b. Nṉgu i-na-phik-ɪtsa maṉgu kwá kadžidzi
  porcupine S-PST-cook-CAUS pumpkins by owl
  ‘The porcupine had the pumpkins cooked by the owl.

(Chichewa, Alsina & Joshi 1991:8)

A similar observation, i.e., that morphological and syntactic constructions serve the same purpose in different languages, is made by Broadwell (2000) for Choctow (Native American, Muskogean) and Oneida (Native American, Iroquoian) directionals. Lomashvili (2011) is concerned with morphological complex predicates in three polysynthetic languages of the South Caucasian (Kartvelian) language family. Following these insights, this thesis does not make a distinction in the analysis of the argument structure and semantics of morphological versus syntactic complex predicates.

As this discussion shows, complex predicates have been studied for a range of different languages with a range of different properties. Of course, complex predicates exist in many more languages, e.g., in Korean as discussed in Sells (1998) and Choi (2005). Most important for the treatment of Murrinh-Patha classifier and lexical stem combinations, however, is the work on similar constructions in other Australian languages. Work on complex predicates in Australian languages comprises so-called coverb constructions that are discussed, among others, by Wilson (1999), Schultze-Berndt (2000), Bowern (2004), Amberber et al. (2007) and Baker & Harvey (2010). (2.9) provides a first example in which the inflecting verb -di ‘COME’ combines with the coverb liri ‘swim’ to form a complex predicate.¹

¹In the glosses for the Australian complex predicates, the inflecting verbs are glossed with small capitals to make the distinction between coverbs and inflecting verbs easily visible. This is adapted from the publications for Murrinh-Patha.
Complex predicates crosslinguistically

(2.9)  
\[ liri-ma \quad ga-di-n \quad lamarra \]
\[ \text{swim-ASP} \quad 3SG-COME-PRS \quad \text{dog} \]
\[ \text{‘the dog is coming swimming’} \quad \quad \text{(Wagiman, Wilson 1999:2)} \]

Amberber et al. (2007) show that the class of coverbs really is a distinct part of speech in Australian languages. The main properties of Australian coverb constructions as well as some dimensions on which Australian coverb constructions differ is discussed in Section 2.4. This section then also summarizes the details of a range of analyses proposed for Australian coverb constructions.

2.2 Light verbs, serial verbs and auxiliaries

This section is concerned with the relationship between complex predicates and other multi-verb constructions.\(^2\) Butt (2003, 2010) points out that constructions have to be evaluated carefully and individually to decide on their status as complex predicates. Verb plus noun complex predicates may have to be tested to decide whether the noun is truly a predicative element or if it is just an argument of the verb. Verb plus verb complex predicates, on the other hand, have to be distinguished from a range of other multi-verb complexes, such as biclausal embedding constructions, auxiliary verb constructions and serial verbs. This section is mainly concerned with the question of how to distinguish complex predicates from auxiliary verb constructions and from serial verbs.

This task is difficult as the surface appearance of these constructions is often quite similar. In all three constructions, usually an element of an open class combines with an element of a closed class, i.e., in complex predicates a light verb combines with a verb, in auxiliary verb constructions an auxiliary combines with a verb and in certain serial verb constructions, one verb comes from a restricted class, too.

Light verbs in complex predicate constructions, auxiliaries and serial verbs are often also very similar semantically. An example is provided in (2.10). All sentences contain an inflected form of ‘stand’ in combination with either an uninflected or inflected “main” verb and it seems that ‘stand’ in these sentences conveys mostly aspectual information. Nevertheless, the constructions have been analyzed (or at least called) differently by different researchers. Lemmens (2005) treats the Dutch example in (2.10a) as an auxiliary construction while Aikhenvald (1999) calls the Tariana example in (2.10b) a serial verb construction and Bowern (2004) uses the Turkmen sentence in (2.10c) as an example for a complex predicate.

\(^2\)An earlier version of this section was published as Seiss (2009).
(2.10) a. Auxiliary Construction

*Ik stond te wachten.*
I stood to wait-INF
‘I was (standing and) waiting.’ (Dutch, Lemmens 2005:184)

b. Serial Verb

*tui-rí-kere na-hwa nema.*
bird-island 3PL-stay 3PL-stand
‘They stayed at the Bird Island for a long time.’
(Tariana, Aikhenvald 1999:480)

c. Complex Predicate

*Ali kitabi okuyup turdu.*
Ali book-ACC read-GER ‘stand’-PST

I do not want to claim here that all these constructions are the same, but I argue that a careful, detailed study to decide on the status of such multi-verbal constructions is needed. Distinguishing between auxiliary, light verb or serial verb constructions goes beyond the merely terminological. A unified terminology enables linguists to compare constructions crosslinguistically and to test analyses proposed for a construction in one language against the same construction in other languages. For example, Baker’s (1989) analysis of serial verbs is often criticized for only accounting for serial verbs which share their objects. Shared objecthood, however, is a defining feature for serial verbs as understood by Baker (1989). His analysis thus cannot be evaluated against serial verbs which do not share their objects.

To avoid such problems, this section proposes a range of crosslinguistically valid criteria to distinguish between auxiliary, light verb and serial verb constructions. For this purpose, it reviews the existing literature on serial verbs, light verbs and auxiliaries and points to problematic aspects of the treatments of these constructions. First, the problem of serial verbs is laid out. The second part of the section is then concerned with the question of auxiliaries, by first looking at the historical development of auxiliaries and then discussing the differences between light verbs and auxiliaries.

2.2.1 The problem of serial verbs

Jarkey (2010) and Appah (2009). In spite of this substantial body of research, still no agreed upon set of defining features of serial verbs has been established. Thus, serial verbs do not seem to be a coherent syntactic class, or, as Crowley (2002:10) put it, “many authors are not fully explicit about what they mean by serial verbs, with some writers simply treating any verb-verb sequence as serial verbs as long as the second verb is not obviously marked as an infinitive”. For a discussion of this problem see for example also Sebba (1987), Seuren (1991), Zwicky (1990) and Lord (1993).

Most importantly, researchers differ in their views on object sharing, switch-subject constructions and shared tense, aspect and polarity features, of which the issue of object sharing is the most controversial. While some researchers, e.g., Stewart (1963), Baker (1989) or Stewart (2001), require objects to be shared, others do not require objects to be shared, e.g., Crowley (2002) or Aikhenvald (1999, 2006). More precisely, most researchers agree on treating sentences like (2.11a) as serial verbs, because `ëv hárê ‘food’, is the object of both verbs. Sentence (2.11b), on the other hand, is a combination of an intransitive and a transitive verb. Thus, the object cannot be shared and some researchers (e.g. Stewart 2001) would not treat this construction as an instance of serial verbs.

\[(2.11)\]
\[\begin{align*}
\text{a. } & \text{Ozo dê `ëv hárê } \text{ rhú } \text{nè Ifuekò} \\
& \text{‘Ozo bought the food and gave it to Ifueko.’} \\
\text{b. } & \text{Uyi hiá lê `ëv hárê} \\
& \text{‘Uyi managed to cook food.’} \\
\text{c. } & \text{Abiêyúwa hiun `ërhán kpáán álímó} \\
& \text{Abieyuwa climbed the tree and plucked an orange.’} \\
\end{align*}\]

(Èdò, Stewart 2001)

(2.11c) is often called ‘covert coordination’ and is still more controversial than (2.11b). Researchers who consider object sharing a defining feature clearly reject this construction as a serial verb. However, others claim that to decide whether (2.11c) is a serial verb construction, semantic and pragmatic features have to be taken into account. Thus, a serial verb can only be used to denote an accepted, although maybe complex, event in a culture. In Alamblak, for example, an action which involves climbing a tree in order to look for insects is a reasonable event, but an action which involves climbing a tree in order to look at the moon is not (Bruce 1988, see also Durie 1997). This meaning cannot be expressed by a serial verb and (2.12b) is thus ungrammatical.
As a result of these differences, different subgroupings have been proposed by different researchers. An early distinction along with a theoretical analysis was proposed by Foley & Olson (1985) who distinguish between nuclear and core layer serialization (see also Crowley 2002), i.e., they distinguish in principle between V and VP serialization. A distinction between covert coordination and serialization of verbs which form a complex event was proposed by Osam (1994), who calls these ‘clause chaining’ and ‘integrated serial verbs’ respectively. This distinction corresponds to what other researchers have called ‘linking type’ and ‘modifying type’ (e.g. Bamgbose 1974).

Aikhenvald (1999, 2006) looks at the problem from a different angle and distinguishes serial verb constructions according to verb classes. In a symmetrical serial verb construction both verbs come from an open verb class while in an asymmetrical serial verb construction one of the verbs comes from a restricted verb class, e.g., from motion or posture verbs. Finally, Andrews & Manning (1999) propose formal analyses for very different serial verbs in Tariana and Misumalpan and discuss different understandings of serial verbs by different researchers.

Although researchers do not agree upon these differences, some properties are shared among all of them. Thus, Bowern (2008a) lists the following concepts as properties of serial verbs in general:

(2.13) Properties of serial verbs (Bowern 2008a)

- the clause contains two (or more) verbs under a single intonation contour
- the verbs must be full lexical verbs which can head simple predicates in their own right
- the verbs share at least one argument
- the verbs behave as a single unit for tense, aspect, and polarity marking

While this set may be the minimal similarities of the constructions called serial verbs in the literature, it would be difficult to find a proper analysis which accounts for all constructions which may fall under this definition. In the same way, these properties make it hard to distinguish serial
verb constructions from auxiliary constructions and complex predicates. As a consequence, serial verbs cannot be compared as a whole class to complex predicates or auxiliaries (see also Beermann & Hellan 2002). Careful language-specific studies are needed to decide whether certain kinds of serial verbs may be auxiliaries or complex predicates. Serial verbs which do not share their object, like causative or aspectual serial verbs, may be complex predicates or auxiliaries. For example, Bodomo (1996), Andrews & Manning (1999) and Nordlinger (2010b) argue to treat serial verbs in specific languages as complex predicates.

Other serial verb constructions may be distinguished from complex predicates and auxiliary constructions, for example symmetrical serial verbs in which both verbs carry their full semantic content, i.e., when they are not “light” verbs. Additionally, morphological marking for tense, person etc., can be on just one, on more or on all verbs in a serial verb construction. On the other hand, morphological marking in complex predicates is usually just on the light verb. Finally, there seems to be a difference in the semantics of many kinds of serial verbs and complex predicates. Thus, verbs in serial verb constructions denote single events which constitute a complex event together while light verbs provide more information about the event of the main verb (Butt 1995) and auxiliaries mainly provide information about tense, aspect and mood.

To sum up, as constructions called serial verbs vary in details such as object sharing etc., they cannot be compared as a whole syntactic class to auxiliaries or light verbs. Common properties of serial verbs as proposed by Bowern (2008a) or Aikhenvald (2006) are useful for an internal typology of serial verbs but do not offer insights into how serial verbs should be distinguished from complex predicates and auxiliaries. To decide whether a given verb series in a specific language may be a light verb or auxiliary, a detailed study of this construction is needed. In the following, I discuss some properties of auxiliaries and light verbs which may help to differentiate between auxiliaries and complex predicates.

### 2.2.2 Auxiliaries and their historical development

When looking at the historical development of auxiliaries, one finds a consensus that auxiliaries may develop from main verbs when they acquire functional properties. A prominent example is constituted by motion and posture verbs as sources for auxiliaries (c.f. Bybee et al. 1994), for example the English *going-to-future* or the Catalan *go-past* (Juge 2006). There also seems to be a consensus that serial verbs can be an intermediate stage on the grammaticalization cline for auxiliaries (Anderson 2006, Heine 1993, Lord 1993, Delancey 1991).

However, researchers do not agree on whether light verbs are an intermediate stage between main verbs and auxiliaries. Roberts & Roussou (2003)
discuss the development of English modal auxiliaries and state that there is some evidence for assuming that the pre-modal verbs, i.e., the verbs which developed into modals, were light verbs. However, they do not discuss this in detail and do not take a definitive view on the matter. Similarly, Hopper & Traugott (1993) follow Hook (1974, 1991) in his proposal that light verbs in Hindi and other Indo-Aryan languages are an intermediate stage between main verbs and auxiliaries. However, Hopper & Traugott (2003) revise this view and state that it is not clear that auxiliaries developed from light verbs. They so acknowledge insights proposed by Butt & Geuder (2003) and Butt & Lahiri (2013) who claim that light verbs do not develop into auxiliaries but are a dead end in the development of verb forms. They show that light verbs in Urdu have been used similarly for thousands of years. Bowern (2008a) agrees with the view that light verbs are not a necessary step for the development from main verbs to auxiliaries but leaves it open if light verbs can develop into other verbal forms or inflections.

In this debate it becomes apparent that a difference in the application of the terms ‘complex predicate’ and ‘light verb’ by different researchers is the, or at least one, reason for their differing views. While for example Butt & Lahiri (2013) have a very clear, narrow definition of light verbs and complex predicates, as in the definition by Butt (1995) given in (2.1), Anderson (2006) includes various syntactic constructions, such as serial verb constructions, verb plus clausal complement sequences, clause-chained or conjunctive sequences, under the label ‘complex predicates’.

Independently of whether light verbs are an intermediate step in the development of auxiliaries, drawing a line between auxiliaries and other verb forms is complicated by the diachronic perspective. In general, we find two major terminological traditions: some researchers (e.g. Kuteva 2001, Lemmens 2005, Anderson 2006) do not make a distinction as to how far a verb has been reanalyzed as an aspect marker. As soon as a verb is used in this way, it is called an auxiliary. Others (e.g. Heine 1993) acknowledge that there is a transition period where the distinction is not clear but for the constructions at the starting and end point of the historic development one can find distinguishing features. For example, in Heine’s (1993) view, an auxiliary has reached its ‘developmental end-point’ when the auxiliary can be used with its corresponding main verb, in sentences like He is going to go to the cinema.

Defining auxiliaries is further complicated by the fact that auxiliaries look very different in different languages. Thus, while most researchers agree that auxiliaries in some way position the event of the main verb in context to the speech or reference time, i.e., they convey information about tense and aspect, other properties of auxiliaries differ from language to language. Thus, in some languages auxiliaries carry all morphological information relating to a predicate such as person, number, tense/aspect/modality, negation marking etc., while in other languages auxiliaries show a reduced verbal behavior.
Connected to this question is the problem whether auxiliaries can combine with inflected main verbs or if they have to carry all inflections themselves.

One example for a combination of an auxiliary with an inflected main verb comes from Urdu. Butt & Rizvi (2010) show that in Urdu, 'be' can be used as an auxiliary marking future tense in combination with main verbs which themselves can be marked in different ways. For example, the main verb can be marked with perfective as well as with imperfective morphology, as can be seen in (2.14).

\[(2.14)\]
\[
a. \quad \text{nadya} \quad \text{a-gi} \quad \text{ho-g-i}
\]
\[
\text{Nadya.FSG.NOM} \quad \text{come-PFV.FSG} \quad \text{be.3SG-FUT-F.SG}
\]
\[
\text{‘Nadya will have arrived/Nadya will probably have arrived.’}
\]
\[
b. \quad \text{nadya} \quad \text{a-ti} \quad \text{ho-g-i}
\]
\[
\text{Nadya.FSG.NOM} \quad \text{come-IMPF.F.SG} \quad \text{be.3SG-FUT-F.SG}
\]
\[
\text{‘Nadya should be arriving.’} \quad \text{(Butt & Rizvi 2010:48)}
\]

One question on which researchers also do not agree is whether the auxiliary may still carry some of its original semantic meaning. Heine (1993), however, points out that this is not a very reliable criterion as even with accepted auxiliaries such as in the English going-to-future, it is not always clear whether \textit{is going to} as in (2.15b) is a grammatical or verbal element.

\[(2.15)\]
\[
a. \text{He is going to town.}
\]
\[
b. \text{He is going to work.}
\]
\[
c. \text{He is going to come.}
\]

Lemmens (2005) looks at aspectual posture verb constructions in Dutch which are used to convey progressive, durative or habitual meaning. Examples of such constructions are given in (2.16).

\[(2.16)\]
\[
\text{Ik zat te lezen / ik stond te wachten / ik lag te slapen.}
\]
\[
\text{I sat to read-INF / I stood to wait-INF / I lay to sleep-INF}
\]
\[
\text{‘I was (sitting and) reading / (standing and) waiting / (lying and) sleeping.’} \quad \text{(Lemmens 2005:184)}
\]

In the examples in (2.16) it can be argued that the meaning of the posture verbs is still important as the meaning of the main predicate fits their meaning. However, these constructions can also be used when the agent’s posture is not an issue, or when the posture denoted by the auxiliary does not correspond to the posture of the main verb, for example as illustrated in (2.17).

\[3\]
I thank Rachel Nordlinger (p.c.) for bringing up this question.
(2.17) *Wat zit ik hier toch rond te lopen?*
   what sit I here (toch) around to walk?
   ‘Why on earth am I walking (around) here?’ (Lemmens 2005:185)

Similar examples are also discussed in Kuteva (2001). These examples make it difficult to draw the line between main verb usage and auxiliary usages as it is not clear whether the specification of the posture in (2.16) plays a more important role than in (2.17). It would thus be very strange to call the posture verb in (2.17) auxiliary but exclude the posture verbs in (2.16) from being an auxiliary in the Dutch verbal system. Thus, in my view an auxiliary can also carry some of the original semantics of the verb it developed from.

Summing up, auxiliaries can develop from main verbs and can mark tense, aspect or modality. They may also carry some of their original semantic meaning and may combine with inflected main verbs. More properties can be set up to distinguish auxiliaries from light verbs, which is discussed in the following subsection.

### 2.2.3 Light verbs vs. auxiliaries

Butt (2010) states that tests to distinguish light verbs from main verbs or auxiliaries differ from language to language. However, there are also some properties which set light verbs apart from auxiliaries crosslinguistically. Butt & Lahiri (2013) name further properties to distinguish light verbs from auxiliaries.

(2.18) Properties of light verbs

- Light verbs are always form identical to the corresponding main verb whereas auxiliaries are usually just form identical at the initial stage of reanalysis from verb to auxiliary (Butt 2010)
- Light verbs always span the entire verbal paradigm (are not restricted to appear with just one tense or aspect form) (Butt & Lahiri 2013)
- Light verbs do not display a defective paradigm (Butt & Lahiri 2013)
- Light verbs exhibit subtle lexical semantic differences in terms of combinatorial possibilities with main verbs, are thus restricted in their combinations. These restrictions are, in contrast to auxiliaries, not necessarily predictable (Butt 2010)

When looking at complex predicates crosslinguistically, further properties of light verbs can be observed which set them apart from auxiliaries, although
Complex predicates crosslinguistically

sometimes a very careful look is needed to distinguish the two constructions. For example, light verbs contribute semantic information about the type of event. This can sometimes include Aktionsart information, which can be confused with aspect, especially if the light verb is encoding telicity/completeness as in (2.19)

(2.19) \textit{anjum}\textsubscript{ne} xat \textit{lk}\textsuperscript{h} \textit{li-yaa}.
\textit{Anjum.F.SG=ERG letter.M.NOM write take-PFV.M.SG}

‘Anjum wrote a letter (completely).’ \textit{(Urdu, Butt 1995:90)}

However, other differences also exist. Thus, light verbs can change the valency of a construction, for example in causative constructions as in (2.20).

The light verb \textit{faire} ‘make’, adds an argument, the causer, to the construction. Auxiliaries are not able to add or reduce arguments. Passive auxiliaries, which may be considered as reducing the arguments at first glance, seem to be very different from light verbs when looked at in more detail. For example, passives do not change the basic argument structure, just its syntactic realization, and the agent can still be expressed as an adjunct.

(2.20) \textit{Jean} a \textit{fait} partir \textit{Marie}.
\textit{Jean has made go Marie}

‘Jean made Marie go.’ \textit{(French, Rosen 1990:37)}

Another property in which light verbs and auxiliaries differ is the ability to assign case. Light verbs may determine case assignment, e.g. in (2.21), the case of the subject depends on the choice for the light verb. Auxiliaries, in contrast, are usually not considered to be able to assign case, but may be sensitive to categories such as unaccusative vs. unergative.

(2.21) a. \textit{ilaa}=ko \textit{k}\textsuperscript{h}aanaa \textit{pasand} \textit{huua}
\textit{Ila=DAT food.NOM like happen.PFV}

‘Ilia liked the food.’ \textit{(Hindi, Mohanan 1997:437)}

b. \textit{ilaa}=ne \textit{k}\textsuperscript{h}aanaa \textit{pasand} \textit{kiyaa}
\textit{Ila=ERG food.NOM like do.PFV}

‘Ilia liked the food.’ \textit{(Hindi, Mohanan 1997:437)}

Finally, light verbs may determine theta-role assignment while auxiliaries cannot. In (2.22), an example from the Australian language Bardi, the light verbs \textit{ma} ‘PUT’ or \textit{ga} ‘CARRY’ result in a different theta-role-assignment when combined with the coverb \textit{abarrabarr}. In (2.22a), there is only one theta-role, a theme. In contrast, in (2.22b), two theta-roles are assigned, an agent and a patient.

(2.22) a. \textit{abarrabarr}-\textit{ma}- ‘to be careless’

b. \textit{abarrabarr}-\textit{ga}- ‘to lead someone astray’ \textit{(Bardi, Bowern 2004)}
More generally, however, it seems to be problematic to distinguish between light verbs in Australian coverb constructions such as in (2.22) above and auxiliary verbs. Schultze-Berndt (2000) makes a distinction between a light verb and an auxiliary use of the verb -yu ‘BE’ in Jaminjung. In (2.23a), -yu ‘BE’ is used as a light verb. In contrast, she suggests that (2.23b), in which the coverb is inflected with the continuous affix -mayan, should be treated as an auxiliary as transitive coverbs cannot combine with intransitive light verbs otherwise in Jaminjung. However, Schultze-Berndt (2000) draws the conclusions only tentatively as she feels that more research is needed on these constructions.

(2.23)  

a. bayirr ga-yu  
   be.supported 3SGS-BE.PRS  
   ‘It is supported.’  
   (Jaminjung, Schultze-Berndt 2000:221)  

b. en janungbari burlug-mayan ga-yu gugu  
   and another drink-CONT 3SGS-BE.PRS water  
   ‘And the other one is drinking water.’  
   (Jaminjung, Schultze-Berndt 2000:195)

Baker & Harvey (2010) discuss Schultze-Berndt’s (2000) proposal and claim that it is not clear whether the coverb burlug ‘drink’ is really transitive in (2.23). Additionally, as was already mentioned by Schultze-Berndt (2000) as well, they point out that coverbs marked with mayan may combine with other light verbs in Jaminjung, such as in (2.24).

(2.24)  

jarr-mayan=biya gan-arra-m=ngarndi ba-ngawu  
   put.down.one-CONT=NOW 3SG>3SG-PUT-PRS=SFOC2 IMP-SEE  
   ‘She keeps putting them down one at a time, look!’  
   (Jaminjung, Schultze-Berndt 2000:131)

Baker & Harvey (2010) conclude from this that it very difficult to distinguish light verb uses and auxiliary uses of inflecting verbs in Jaminjung and in Australian languages more generally. Chapter 3 will deal with similar constructions in Murrinh-Patha and also tentatively conclude that no clear line can be drawn between Murrinh-Patha classifier stems used as light verbs and auxiliaries.

2.3 Analyses of complex predicates

A wide range of different analyses have been proposed for complex predicates across various languages and theoretical frameworks. Complex predicates pose problems for grammatical theories as two or more syntactically independent elements behave like a single predicate. They challenge theories
about the relationship between morphology and syntax on the one hand and the relation between syntactic structure and argument and/or semantic structure on the other hand.

As Butt (1995) points out, the first approaches to complex predicate formation were mainly concerned with the question of monoclausality, e.g., the phrasal account of head-to-head movement proposed by Baker (1988) within Government and Binding or the clause union analysis (Aissen & Perlmutter 1983) within Relational Grammar. However, as Butt (1995) further lays out, these approaches could not distinguish very well between complex predicates and other multi-verb constructions. Later approaches have thus tended to cast their analyses in terms of argument and semantic structure. Although similar in their main underlying idea, these analyses still vary to a great extent with regard to the semantic content that the two elements contribute and with regard to the frameworks chosen for the formal modeling of complex predicate formation.

An early approach to complex predicate formation making use of argument structure was proposed by Grimshaw & Mester (1988) who analyze Japanese *suru* ‘do’ as a light verb. An example of the construction has already been given above and is repeated in (2.25). Grimshaw & Mester (1988) propose the notion of *argument transfer* for this construction: the light verb *suru* ‘do’ does not provide any arguments of its own, but receives the arguments of the noun it combines with. A light verb in their view is thus very similar to an auxiliary or copula and does not contribute to the semantics of the construction. However, more recent approaches (Isoda 1991, Butt 1995, Matsumoto 1996, Miyamoto 1999) argue that *saru*’s argument structure is not completely empty, as case marking facts indicate.

(2.25) a. John-wa Mary-ni hanashi-o shita
   John-TOP Mary-to talk-ACC suru
   ‘John talked to Mary.’
   a. John-wa Tookyoo-kara shuppatsu-o shita
   John-TOP Tokyo-from depart-ACC suru
   ‘John departed from Tokyo.’

   (Japanese, Grimshaw & Mester 1988:207)

Grimshaw & Mester (1988) use thematic roles to model the argument structure operations triggered in Japanese complex predicate formation. Other approaches to complex predicate formation also use different models of argument structure for their analyses. For example, Alsina (1996, 1997) uses Dowty’s (1991) Proto-Roles to model causatives in Catalan and Chichewa. He is concerned with examples such as (2.26).

(2.26) a. Afisi a-na-sék-a
   2P-hyenas 2P-S-PH-laugh-FV
   ‘The hyenas laughed.’
b. *Njóvu i-na-sek-éts-a afisi*
   IX elephant IX S-PA-laugh-CAUS-FV II hyenas
   ‘The elephant made the hyenas laugh.’

(Chichewa, Alsina 1997:209)

To model complex predicate formation, Alsina (1997) assumes the argument structure as specified in (2.27) for the verb *sék-a* ‘laugh’ and the causativizing morpheme *its-a*. (2.27a) specifies that the verb *sek-a* ‘laugh’ needs a Proto-Agent (P-A) as its argument. The argument structure of the causativizer *its-a* in (2.27b) is more complex as it requires a Proto-Agent as well as a Proto-Patient (P-P) and additionally requires that a predicate (P*) combines with it.

(2.27)  

a. \[ \text{ag} \]

\[
\text{sek-a: ‘laugh} < [ P-A ] > ‘ \quad \text{ (Alsina 1997:206)}
\]

b. \[ \text{ag pt} \]

\[
\text{its-a: ‘cause} < [ P-A ] [ P-P ] P^* < \ldots [ ] \ldots > \quad \text{ (Alsina 1997:211)}
\]

In complex predicate formation, the argument structure of the verb is inserted into the argument structure of the causativizer for the placeholder P* and the object of the causativizer is coindexed with the subject of the verb. This can be seen in (2.28).

(2.28)  

\[ \text{ag pt ag} \]

\[
\text{sek-éts-a: ‘cause} < [ P-A ] [ P-P ] \text{laugh} < [ P-A ] >> ‘
\]

\[
\text{SUBJ OBJ} \quad \text{ (Alsina 1997:212)}
\]

Alsina’s (1997) approach to complex predicate formation demonstrates nicely the main underlying idea of many approaches to complex predicate formation: that both elements of the complex predicate have an argument structure that needs to combine in a certain way. Usually, the argument structure of the light verb, i.e., the causativizer in Alsina’s (1997) case, is deficient and so triggers the combination with another predicational element.

In (2.28), the patient argument of the causativizer is coindexed with the agent argument of the verb which is then mapped onto the object. This violates the so-called Function-Argument Bi-uniqueness principle normally assumed in LFG (Bresnan 2001:311) or the Uniformity of Theta Assignment Hypothesis of GB (Baker 1988). These principles assume that each thematic role should be mapped onto one single grammatical function and vice versa.
As will be seen in the following, complex predicates in general seem to question these principles.

Alsina (1997) uses Dowty’s (1991) Proto-Roles. However, a wide range of other approaches to argument structure have been used as well. Mohanan (1997), for example, uses a combination of argument structure and semantic structure to model complex predicate formation in Hindi noun plus verb complex predicates. She argues for factoring apart the information normally associated with thematic roles into argument structure and semantic structure. In her view, argument structure is a pure representation of argument slots or placeholders while semantic structure should incorporate finer-grained semantic representations. She uses mnemonic labels such as *truster* and *trusted* to model the semantic structure for the purpose of the analysis, but states that more abstract structures would be needed instead. What she makes clear, however, is that more semantic information is needed in complex predicate formation than what can be encoded with traditional thematic role labels.

Similarly, Butt (1995) requires more information than encoded with thematic roles in her analysis of Urdu complex predicates. She focuses on the two verb plus verb complex predicate constructions illustrated in (2.6) above. (2.29) provides additional examples. In (2.29a), the verbs *de* ‘let’ and *banaa* ‘make’ combine to form a permissive complex predicate while (2.29b) is an example of an aspectual complex predicate involving *le* ‘take’ and *banaa* ‘make’.

(2.29) a. *anjum=ne saddaf=ko haar*
   Anjum.F=ERG Saddaf.F=DAT necklace.M.NOM
   *banaa-ne di-yaa*
   make-Inf.OBL give-PFV.M.SG
   ‘Anjum let Saddaf make a necklace.’ (Urdu, Butt 1995:35)

b. *anjum=ne haar banaa*
   Anjum.F=ERG necklace.M.NOM make-Inf.OBL
   *li-yaa*
   take-PFV.M.SG
   ‘Anjum completed making the necklace.’
   (Urdu, Butt 1995:152)

Butt (1995) shows that in both of these constructions, the overall structure is monoclausal and the argument structure is jointly determined by the two verbs. For the formal modeling of the argument structure, Butt (1995) adopts Jackendoff’s Lexical Conceptual Structures, which for Jackendoff (1990) belong to the level of semantics that can be directly linked to pragmatics. Butt (1995), however, uses lcss as an elaborate argument structure which distinguishes her approach from the approach proposed by Mohanan (1997) considerably.
The underlying idea of the formalism is that both verbs, i.e., the light verb and the main verb, can be given an LCS analysis and these LCSs can be merged in complex predicate formation. In order to form a coherent complex predicate, the LCS of the light verb and the LCS of the main verb need to be compatible.

For the modeling of the combinatory possibilities of light and main verbs, Butt (1995) adapts the formalism of LCSs to the requirements set by her data. For example, Jackendoff (1990) proposes two tiers for his representation of LCSs, the Thematic Tier and the Action Tier. While the Thematic Tier represents the meaning of the verb, the Action Tier encodes actor, patient and beneficiary relations. The two representations are then linked by coindexing. Butt (1995) adds a third tier to this representation, which she calls Aspect Tier. She needs this tier to model the different aspectual information that both light verbs and main verbs carry and that restrict the possible combinations of these. She proposes a function ASP with three slots in which “the first slot represents the starting point of an event, the second slot the duration, and the third slot the end point” (Butt 1995:142). The slots can be underspecified (0) or positively (1) or negatively (0) specified.

In their LCS representation, the light verbs, such as de ‘give’ and le ‘take’ in (2.29), differ from their corresponding main verb usage, as they contain a transparent event. Butt (1995) leaves the exact notion of a transparent event unclear, but in her view a transparent event is an argument which cannot stand on its own and therefore has to trigger complex predicate formation. She marks transparent events with \{ \} in LCSSs.

When the LCSs are merged, either event fusion or argument fusion takes place, depending on the light verb. In the case of the aspectual complex predicates, event fusion ensures that the LCSs of the light verb and the main verb are merged. The LCS of the light verb le ‘take’ is given in (2.30). The LCS does not contain a Thematic Tier, as the light verb is bleached of its semantic content. However, the LCS does contain an Action Tier which specifies that the argument of the light verb is positively specified for conscious choice [+cc] and an Aspectual Tier which is positively specified for completion.

(2.30) \[ \begin{align*}
&\text{le ‘take’} \\
&\left\{ \begin{array}{l}
\text{AFF}_{+cc}(\emptyset) \\
\text{ASP}(+1)
\end{array} \right\}_{E_T}
\end{align*} \]

In complex predicate formation, the LCS of the main verb and the aspectual light verb is merged. This process is called event fusion. In this process, the specifications of the LCSs need to be compatible, as the LCSs cannot be combined otherwise. That is, the system predicts that the light verb
le ‘take’ cannot combine with a main verb that is negatively specified for completion or negatively specified for conscious choice. The complete LCS of the complex predicate is displayed in (2.31).

\[
\text{(2.31) } \begin{bmatrix}
\text{CS}\left(\left[\alpha\right] , \text{BE}[\text{necklace}]\right) \\
\text{AFF}_{cc}\left(\left[\text{Anjum}\right]^{\alpha}, \right) \\
\text{ASP}(\_ - 1) \\
\end{bmatrix} E
\]

In contrast to aspectual complex predicates which involve event fusion, Butt (1995) proposes that the permissive involves argument fusion. This is the case because the light verb \textit{de} ‘let’ is not as bleached as the aspectual light verbs and still retains some of its structure from its main verb use ‘give’. The LCS for the light use of \textit{de} is given in (2.32). It is the same as the LCS of the main use of \textit{de} except for the fact that the argument of the function \textit{GO}_{pos} is a transparent event, i.e., this event has to be filled in by the main verb.

\[
\text{(2.32) } \begin{bmatrix}
\text{de ‘let’} \\
\text{CS}\left(\left[\alpha\right] , \text{GO}_{pos}\left(\left\{ E_T, \text{TO}\left(\_\right) \right\}\right)\right) \\
\text{AFF}_{cc}\left(\left[\_\right]^{\alpha}, \right) \\
\text{ASP}(\_ - 1) \\
\end{bmatrix} E
\]

(2.33) presents the LCS of the whole complex predicate. The LCS of \textit{banaane} ‘make’ has been inserted into the transparent event specified by the light verb. The process of argument fusion then ensures that the arguments of the light verb and the main verb are coindexed. In (2.33) this can be seen by the coindexation with \beta of Saddaf with the first argument of the Action Tier of the main verb.

\[
\text{(2.33) } \begin{bmatrix}
\text{banaane diyaa ‘let make’} \\
\text{CS}\left(\left[\alpha\right], \text{GO}_{pos}\left(\left\{\left(\text{CS}\left(\left[\gamma\right], \text{BE}[\text{necklace}]\right)\right) \text{AFF}_{cc}\left(\left[\_\right]^{\beta}, \right) \text{ASP}(\_ - 1) \text{E}_{T}, \text{TO}\left[\text{Saddaf}\right]\beta\right)\right)\right) \\
\text{AFF}_{cc}\left(\left[\text{Anjum}\right]^{\alpha}, \right) \\
\text{ASP}(\_ - 1) \\
\end{bmatrix} E
\]

As the whole LCS of the main verb is inserted into the transparent event of the light verb, the system predicts that all main verbs should be able to combine with the light verb \textit{de} ‘let’ in permissive complex predicates. As will be seen in the following section, this is not the case for complex
predicate combinations in other languages in which the semantics of the involved verbs plays an even bigger role in the selection of the complex predicate combinations. Nevertheless, the basic underlying idea of modeling the combinatory possibilities of complex predicates with LCSs has been adopted for a range of different complex predicate constructions such as Romance causatives (by Butt 1995 herself and Andrews & Manning 1999), Japanese suru ‘do’ (by Butt 1995 herself), Choctaw directionals (Broadwell 2000), Australian coverb constructions (Wilson 1999, Wilson 2006, Baker & Harvey 2010) or serial verb constructions crosslinguistically (Durie 1997) or in the Australian language Wambaya (Nordlinger 2010b). Sections 2.4.1 and 2.4.2 present Wilson’s (1999) account of Wagiman complex predicates and Baker and Harvey’s (2010) account of coverb constructions in more detail.

Butt’s (1995) analysis is cast within LFG for which she proposes a new linking theory that maps the arguments defined in the argument structure onto the grammatical relations. In contrast to standard views in LFG, this linking does not happen in the lexicon, as the argument structure is determined by two separate lexical items in syntax.

A different line of modeling the argument structure is pursued by a range of work cast within First Phase Syntax. Hale & Keyser (2002) assume that argument structure is part of syntax rather than part of the lexicon. They propose that all verbs have a complex underlying structure and therefore represent this complex structure for both complex and simple predicates. In their approach, they treat English simple verbs and complex predicates in various languages alike. Their approach has also been applied to, e.g., Persian complex predicates, for example by Megerdoomian (2001) and Folli et al. (2005).

In the same line of research, Ramchand (2008b), as well as Butt & Ramchand (2005), model Hindi/Urdu aspectual complex predicates. Ramchand (2008b) proposes that what is traditionally assumed to be a V should be decomposed in a range of other phrases. More precisely, she proposes to decompose V into an initP (for initiation), a procP (for process) and a resP (for result). The exact specifications of the phrases are given in (2.34), taken from Ramchand (2008b:40).

(2.34) a. initP introduces the causation event and licenses the external argument (‘subject’ of cause = INITIATOR)

b. procP specifies the nature of the change or process and licenses the entity undergoing change or process (‘subject’ of process = UNDERGOER)

c. resP gives the ‘telos’ or ‘result state’ of the event and licenses the entity that comes to hold the result state (‘subject’ of result = RESULTEE)

For these phrases, she argues that init selects for procP and proc selects for
resP. In this way, a process always needs an initiator and a result always needs a process. The three roles INITIATOR, UNDERGOER and RESULTEE can be mapped onto the arguments of verbs in different ways. The roles are composite in the way that two or even three roles can be realized as one argument of the verb. For example, she argues that the verb arrive is an intransitive verb whose subject fulfills all three roles, the INITIATOR, UNDERGOER and RESULTEE. This system thus allows her to model different English verb classes based on the different roles that the arguments of the verb have.

She also applies this system to complex predicate formation in Hindi/Urdu. The underlying idea is that the elements building the complex predicate jointly contribute to the composition of V by each specifying a certain aspect of the phrase structure. (2.35) presents an example of the Hindi/Urdu aspectual complex predicate that she discusses.

\[(2.35)\] \[nadya=ne xat lik\hat{h} li-yaa\]
\[\text{Nadya.F=ERG letter.M.NOM write take-PVF.M.SG}\]
\[\text{‘Nadya wrote the letter (completely).’}\]

\((\text{Hindi/Urdu, Ramchand 2008b:146})\)

The analysis of (2.35) as proposed by Ramchand (2008b) is illustrated in (2.36). The main verb lik ‘write’ specifies the result state and therefore contributes the head of resP. In contrast, the light verb ‘take’ is identified as the head of initP and procP. As Ramchand (2008b) points out, it is the light verb that selects for the resP, i.e., it is the light verb that specifies the completive information, but it is the main verb that determines the result state, e.g., of something being written.

\[(2.36)\] 
\[
\text{DP} \quad \text{initP} \quad \text{procP} \quad \text{init} \quad \text{take} \\
\text{Nadya} \quad \text{DP} \quad \text{DP_i} \quad \text{resP} \quad \text{proc} \quad \langle\text{take}\rangle \\
\text{letter} \quad \text{res} \quad \text{written} \quad \langle\text{letter}\rangle \]
Ramchand’s (2008) approach has also been applied to Persian by Pantcheva (2009) and Telugu by Balusu (2012).

The approaches to complex predicate formation on the level of argument structure and semantics assume a rather static analysis. However, the interpretation of a complex predicate might also depend on the context and pragmatic principles. Tantos (2008) is concerned with constructions involving the light verb *have* in English such as in (2.37), which is ambiguous in that the subject can be the causer, i.e., John made his class walk out on him, or the experiencer. Which reading is actually involved is determined by the context. For the modeling of this context-specificity, Tantos (2008) uses Segmented Discourse Representation Theory (Asher & Lascarides 2003).

(2.37) John had his class walk out on him. (Tantos 2008:1)

Caudal et al. (2012) propose a context-sensitive modeling of verb formation patterns in the Australian language Panyjima, using insights from Asher’s (2011) Type Composition Logic.

Thus, a range of very diverse analyses have been proposed for complex predicates crosslinguistically. Similarly, analyses for Australian coverb constructions differ in their frameworks and assumptions as well. Wilson’s (1999) analysis of Wagiman complex predicates builds on work by Butt (1995) and Andrews & Manning (1999) within LFG, modeling the argument structure with LCSs. In their treatment of Australian coverb constructions and crosslinguistic serial verbs, Baker & Harvey (2010) model the argument structure with LCSs as well, but are not explicit about the framework they adopt. Schultze-Berndt (2000) proposes a construction-based analysis for a wide range of data from Jaminjung, a Northern Australian language from the Victoria River area. Bowern (2004), on the other hand, takes on a historical perspective on coverb plus inflecting verb constructions in Bardi, a Nyulnyulan language from the North-Western Australian coast. Finally, McGregor (2002) provides another perspective in analyzing coverb plus inflecting verb constructions in Australian languages. In his view, inflecting verbs act as classifiers for the group of coverbs. In the following section, these analyses of Australian coverb constructions, along with similarities and differences in the various Australian languages, are discussed in greater detail.

### 2.4 Australian coverb constructions

As was discussed in Chapter 1, most Murrinh-Patha verbs have a bipartite verbal structure in which the lexical semantic meaning is determined by two elements, the classifier stem and the lexical stem. The classifier stems inflect for subject person and number as well as for tense while the lexical stems
are non-inflecting. Similar constructions can be found in a range of different Australian languages, actually in most languages of North-Central and North-Western Australia (Baker & Harvey 2010). These constructions can be either morphological or syntactic combinations of two elements. They share the property that an inflecting item combines with a non-inflecting lexical item to form the verbal complex. However, the details of the properties as well as the terminology concerned with the constructions exhibit considerable differences.

Two main distinctions in the terminology can be found. Some approaches treat, or at least call, the inflecting lexical item “auxiliary” and the non-inflecting item “(main) verb”, e.g., Dixon (1972) and Tryon (1974), while others call the inflecting element “verb” and consider the non-inflecting element as a special part of speech, e.g. Nash (1986).

Within these two broad approaches, still different terminology is used. If the non-inflecting element is treated as a separate part of speech, it is referred to as coverb (Schultze-Berndt 2000, Wilson 1999, Amberber et al. 2007), preverb (Nash 1986, Simpson 1991, Bowern 2004), verb particle (Merlan 1994) or participle (Cook 1987). However, the terms coverb and preverb seem to have become the most commonly used. In this tradition, the other element is then usually called inflecting verb, non-finite verb or simply just verb.

If the inflecting element is called auxiliary and the non-inflecting part verb, the combination is usually considered to be an instance of classification in which the auxiliary classifies the verb into these different conjugation classes (Capell 1976, 1979). McGregor (1990, 2002) for this reason calls the inflecting element a classifier.

In addition to this distinction, the terminology also often differs for syntactic versus morphological combinations of the inflecting and non-inflecting elements. In this work, I use the terms classifier and lexical stem for morphological complex predicates, as these terms are usually used in discussing Murrinh-Patha complex predicates. The terms inflecting verb and coverb are used for syntactic complex predicates and the combination of both elements is referred to as the coverb construction.

Apart from the terminological differences, real differences in the properties of these bipartite constructions also exist. In the following, I discuss some of the properties of Australian coverb constructions. For the purpose of this dissertation, the discussion is restricted to the comparison of coverb constructions, i.e., combinations of inflecting and non-inflecting elements, across different Australian languages. This excludes coverb constructions in languages outside of Australia, which can be found, according to Amberber et al. (2007), in Ethio-Semitic, Cushitic and Omotic languages. This also excludes other types of complex predicate formation in Australian languages, e.g., verb compounding in which two verbs combine to form a main verb such as, e.g., in Alyawarra (Yallop 1977) or the verbalizing morphology described
by Dras et al. (2012) for Arrernte or Caudal et al. (2012) for Panyjima. While there might be diachronic links between these systems and the coverb construction, as discussed in e.g., Dixon (2002), Schultze-Berndt (2003) and Caudal et al. (2013b), synchronically these systems are very different from the Murrinh-Patha construction.

The coverb constructions differ in various dimensions across different Australian languages. One considerable difference is between languages which have a large inventory of simple verbs and use complex predicates additionally. On the other end of the extreme are languages which almost always use complex predicates. Languages also vary with respect to the usage of coverbs, e.g., whether coverbs can be used without inflecting verbs. A further difference regards the linear ordering of the two lexical items. These differences among Australian languages and what properties can be found in Murrinh-Patha complex predicates, are discussed in the following.

Capell (1976) attempts a first classification into various types of coverb constructions, and he proposes that the coverb construction in the Daly river languages differs quite considerably from other coverb constructions. Tryon (1976) looks at the Daly river languages in more detail, discussing similarities and differences between these languages. As Murrinh-Patha was considered a language isolate at that time, Murrinh-Patha is not considered in this study. However, Tryon (1976) considers Ngangikurrungurr, one dialect of Ngan’gityemerri (Reid & McTaggart 2008), which has been shown to be closely related to Murrinh-Patha (Green 2003). The observation that the coverb construction in the Daly river languages differ in certain important aspects from other Australian coverb constructions can thus be extended to Murrinh-Patha as well.

Murrinh-Patha has 38 classifier stems of which only a small subclass can be used without lexical stems. Tryon (1976) claims that in Daly river languages generally, only intransitive inflecting verbs can be used as sole verbal predicate, while transitive inflecting verbs cannot. While this is not completely true in Murrinh-Patha, e.g., the classifier stem take(22) can be used without a lexical stem, it is true that not all classifier stems can function as the sole verbal predicate. In this property, Murrinh-Patha differs from most other Australian languages, as classifier stems or inflecting verbs can most often occur as sole verbal predicates.

In Murrinh-Patha, all classifier stems but say/do(34) take part in complex predicate formation, which is also not the case for most other Australian languages. For example, Bowern (2004) states that Bardi has around 250 inflecting verbs of which 39 can be used in complex predicates, and from these only 12 seem to be productive in the sense that they can combine with more than a few coverbs. In Mawng, only 44 of 600 inflecting verbs are attested in coverb constructions (Singer 2005). From the probably 45 inflecting verbs in Wagiman, only 13 are usually used in complex predicates (Wilson 1999). Most similar to the Murrinh-Patha and Ngangi’tyemeri system in this re-
spect seems to be Jaminjung, which has around 35 different inflecting verbs of which probably 29 occur in complex predicates (Schultze-Berndt 2000).

Australian languages also differ in the semantic range from which inflecting verbs are drawn. As Baker & Harvey (2010) point out, the inventories of Wagiman or Jaminjung are quite similar to other light verb inventories crosslinguistically. They claim that the light verb inventory in Ngan’gityemerri, and in its closest neighbors, among them Murrinh-Patha, differs from the crosslinguistically known inventory in some very important aspects.

Firstly, they point out that verbs of posture constitute a large class of the intransitive inflecting verbs. In Ngan’gityemerri, 4 different posture verbs (sit, lie, stand, perch) can be found, while Jaminjung has no posture verb in its light verb inventory and the only posture verb found in Wagiman, ‘stand’, is only very rarely used. Additionally, Baker & Harvey (2010) point out that these posture verbs, along with motion verbs, are used in constructions of ‘associated stance’ and ‘associated motion’ in Ngan’gityemerri such as in (2.38). They claim that this meaning is usually not conveyed with coverb constructions crosslinguistically.

\[(2.38)\]
\[\begin{align*}
\text{a. } ngirim-fifi & \quad 1SGS.SIT.PRS-smoke \\
& \quad 'I am sitting smoking.' \quad \text{(Ngan’gityemerri, Reid 2000:338)} \\
\text{b. } ngirribem-fifi & \quad 1SGS.STAND.PRS-smoke \\
& \quad 'I am standing smoking.' \quad \text{(Ngan’gityemerri, Reid 2000:338)}
\end{align*}\]

Although Baker & Harvey (2010) may be right in their observation about posture verbs usually not being part of the inventory of light verbs crosslinguistically, there is some evidence that at least the Wagiman system used to be not as dissimilar from the Ngan’gityemerri system. As Baker & Harvey (2010) themselves point out, the two light verbs meaning ‘be’ in Wagiman historically meant ‘lie’ and ‘sit’. To what extent the Ngan’gityemerri system (and for the same reason also its neighbor Murrinh-Patha) just represents an earlier stage of the inventory of e.g., Wagiman, still needs to be resolved.

---

\(^4\)The examples from Ngan’gityemerri are taken from three main sources (Reid 2000, 2002, Reid & McTaggart 2008) which use different glosses. To ensure compatibility between the examples in this work, a unified gloss has been used. This included adding the tense information for the classifier stem for the data taken from Reid (2000). Reid (2000) splits the classifier stem into two morphemes to encode subject information in a separate morpheme. However, as this is not done in Reid (2002) and Reid & McTaggart (2008), the classifier stem is not split into two morphemes here. Reid & McTaggart (2008) do not provide glosses for their dictionary examples, so these examples have been glossed by the author herself. Reid & McTaggart (2008) also spell the classifier and lexical stem in separate words. This has been changed to ensure a unified treatment of all Ngan’gityemerri examples.
Baker & Harvey (2010) further point out that the Ngan’gityemerri system is also singular in having reflexive counterparts to the transitive classifier stems. These reflexive classifier stems are formally unrelated to their corresponding counterparts. An example is provided in (2.39). This contrast can be found in Murrinh-Patha as well.

(2.39)  a. **ngeriny-∅-syirr**
        1SGA.HANDS.PST-3SG.DO-scratch
        ‘I scratched her.’ (Ngan’gityemerri, Reid 2000:348)

       a. **ngemeny-syirr**
        1SGA.HANDS.REFL.PST-scratch
        ‘I scratched myself.’ (Ngan’gityemerri, Reid 2000:348)

While this is a true distinction between the inflecting verb inventories of Ngan’gityemerri on the one hand and most other inflecting verb inventories in Australian languages, I do not consider the inflecting verb inventories of Ngan’gityemerri and Murrinh-Patha as so fundamentally different as Baker & Harvey (2010) do. Some transitive light verbs in Ngan’gityemerri and Murrinh-Patha show some significant similarities with light verbs in other languages, e.g. in Jaminjung. POKE, for example, can be found in the inventory of Murrinh-Patha and Jaminjung. Similarly, both languages make use of an inflecting verb or classifier stem denoting actions that are carried out by the feet, glossed as KICK in Jaminjung and FEET in Murrinh-Patha. Details of the inventory of the Murrinh-Patha classifier stems are presented in Chapter 3. The Ngan’gityemerri system is presented in Section 2.4.5.

Further differences between coverb constructions in Australian languages can also be found with respect to the coverbs or lexical stems. In some Australian languages, the coverb can be used without an inflecting verb. This is the case for example for Wagiman (Wilson 1999), Wardaman (Merlan 1994) and Jaminjung (Schultze-Berndt 2000). Naturally, this makes establishing the semantic meaning of the coverb easier. Wagiman is an extreme example in that coverbs can be used as main verbs – Wilson (1999) speaks of zero-derivation from a coverb to a main verb – and then receive the required morphological marking. As Schultze-Berndt (2000) points out, the use of coverbs as sole verbal predicates is more restricted in Jaminjung, but still the coverb can form the sole verbal predicate in subordinate clauses and the coverb can also be nominalized.

In Murrinh-Patha, the lexical stem can never occur on its own as a predicate. Some of the lexical stems, however, can be linked to other parts of speech, especially nouns and adjectives. One example is bebe ‘vomit’, which according to Street (1989) can be used e.g., with the classifier stems SIT(1) or as a noun. Similarly, birlbirl can be used as an adjective meaning ‘watchful’ and as a lexical stem in combination with e.g., the classifier stem STAND(3) to form a predicate meaning ‘keep watch’ (Street 1989). Despite
Complex predicates crosslinguistically

this syncretism for selected lexical items, lexical stems in Murrinh-Patha cannot be used productively without classifier stems.

Finally, Australian languages vary in the combinatory possibilities of inflecting verbs and coverbs. This of course depends partly on the inflecting verb inventory, as a different inventory of classifying elements usually leads to a different classification system. However, differences in the combinatory possibilities which cannot be explained by the varying inventories of light verbs can also be found. This especially concerns argument structure requirements. As Bowern (2004) points out, Bardi and the other Nyulnylan languages differ from many other Australian languages in that transitive inflecting verbs can form intransitive complex predicates. This phenomenon can be found in Murrinh-Patha as well, but is not found, e.g., in Wagiman (Wilson 1999).

As Baker & Harvey (2010) point out, in a range of Australian languages, e.g. in Kamu, Malak-Malak and Matngele, complex predicates may actually involve more than one coverb. An example of such a construction is provided in (2.40) in which the coverbs dal ‘poken’ and ngak ‘eat’ combine with the inflecting verb go.

(2.40) *dal-ngak*-*ma=*gu-*yang

| poke-eat-IMPF=3SGS-GO.PRS |

‘(The bird) is pecking (at the food).’

lit: ‘(The bird) is poking, eating (the food).’

(Kamu, Baker & Harvey 2010:43)

Similarly, multiple classifier stems can be combined with one lexical stem in some languages, among them Ngan’gityemerri and Murrinh-Patha. In the Ngan’gityemerri example in (2.41), the classifier stems lie and stand serialize onto a combination of a classifier and lexical stem to encode the position of the subject. As Reid (2000) points out, the meaning of the serialized classifier stem most often denotes aspectual information instead of stance.

(2.41) a. *Yawul* karrityinmade *ngebem-wurity=*ngi*bem*

| spear | bent |

1sgS.BASH.PRS-fix=1sgS.LIE.PRS

| ttyatma |

straight

‘I’m lying straightening this bent spear.’

b. *Yawul* karrityinmade *ngebem-wurity=*ngirribem

| spear | bent |

1sgS.BASH.PRS-fix=1sgS.STAND.PRS

| ttyatma |

straight

‘I’m standing straightening this bent spear.’

(Ngan’gityemerri, Reid 2002:258)
After this general overview over Australian coverb constructions, the remainder of this section looks at detailed treatments of Australian coverb constructions. Section 2.4.1 presents Wilson (1999)’s analysis of Wagiman complex predicates which uses ideas put forth by Butt (1995). Similarly, Baker & Harvey (2010) build on insights from Butt (1995) and Wilson (1999) in their analysis of coverb constructions across various Australian languages. Their approach is summarized in 2.4.2. Schultze-Berndt’s (2000) analysis of Jaminjung complex predicates is the topic of Section 2.4.3 while Bowern’s (2004) analysis of Bardi complex predicates is dealt with in Section 2.4.4. Murrinh-Patha is closely related to Ngan’gityemerri, and a detailed look at the alternation patterns found in Ngan’gityemerri can help understand the Murrinh-Patha system. Section 2.4.5 therefore discusses the Ngan’gityemerri alternation patterns as laid out in Reid (2000). Finally, the verbal classification approach by McGregor (2002) is discussed in Section 2.4.6.

2.4.1 Wagiman complex predicates (Wilson 1999)

Wilson (1999) is concerned with complex predicates in the Northern Australian language Wagiman, which is a non-Pama-Nyungan language spoken south of Darwin. In Wagiman, the two verbal elements are two separate morphological words and are called inflecting verbs and coverbs by Wilson (1999). This section first discusses the empirical properties of the Wagiman complex predicates briefly. It then sketches the LCS analysis proposed by Wilson (1999).

For his discussion of the combinatory possibilities of Wagiman inflecting verbs and coverbs, Wilson (1999) groups the coverbs into different semantic classes, e.g., state coverbs, coverbs of change of state, coverbs of motion, coverbs of communication, coverbs of emotion, coverbs of impact and violence and adverbial coverbs. Wagiman coverbs are easily grouped into these classes as coverbs can be used as independent predicates in Wagiman. In this use, they inflect for tense and aspect. As Wilson (1999) points out, this is not found in any other language of the Northern Territory. Additionally, coverbs can take the derivational morpheme -y to form a nominal. In this case, the coverbs can take nominal case marking and usually function as clausal adjuncts. The fact that coverbs may be used without inflecting verbs in these constructions makes it relatively easy to determine the basic semantic meaning and even the argument structure of the coverbs.

As Wilson (1999) points out, coverbs of state, change of state and motion are usually intransitive and mostly combine with intransitive inflecting verbs. If they are combined with transitive inflecting verbs, the combination is a causative construction. Apart from these similarities, however, also differences between the classes of coverbs exist.

State coverbs usually combine with the inflecting verbs -yu- ‘BE’, -ni- ‘BE’
Complex predicates crosslinguistically

and -yobe- ‘STAY’, as is illustrated in (2.42a). They may also combine with -ya(ma)- ‘SAY, DO, BECOME’ in its ‘become’ reading to form inchoatives, as in (2.42b).\(^5\) In contrast, change of state coverbs have to combine with -ya(ma)- ‘SAY, DO, BECOME’ as in (2.42c), they cannot combine with the stative inflecting verbs -yu- ‘BE’, -ni- ‘BE’ and -yobe- ‘STAY’, as in (2.42d).

(2.42) a. ga-yu
    3SG.BE.PRS
    3sg.
    ‘that baby is asleep.’
    Wagiman, Wilson 1999:150

b. desh ngaha-ny-ma
    be.white.PFV 1SG.BECOME-PPFV-DS
    ‘my face went white (in fear).’
    Wagiman, Wilson 1999:152

c. wal yaha-ny
    3SG.BECOME-PPFV
    ‘her body has grown’
    Wagiman, Wilson 1999:152

d. *bort-da ga-yu
    die-ASP 3SG.BE.PRES
    ‘he is dead.’
    Wagiman, Wilson 1999:151

For these classes of coverbs, the transitivity of the complex predicate is determined by the transitivity of the inflecting verb. If intransitive coverbs combine with intransitive inflecting verbs, the complex predicate is intransitive. If intransitive coverbs combine with transitive inflecting verbs, the complex predicate is transitive, because a causer is added to the argument structure.

The situation is different for other coverb classes. Combinations of e.g. intransitive stative inflecting verbs and coverbs of communication may be either intransitive or transitive. Wilson (1999) points out that some coverbs of communication can take oblique arguments, as can be seen in (2.43). In (2.43a), the coverb nangay ‘wave’ is used in an intransitive complex predicate with the inflecting verb -yobe- ‘STAY’, denoting an activity. In (2.43b) in contrast, the complex predicate formed by nangay ‘wave’ and the stative inflecting verb -yu- ‘BE’ takes an oblique argument and is thus transitive.

(2.43) a. nga-nyar-ma-n
    1SG-be.tired-VERB-PRS
    3SG.OBL-DS // 3SG-STAY.FUT
    wave-ASp
    ‘I am tired of him, let him keep on waving.’
    Wagiman, Wilson 1999:113

\(^5\)// marks intonation boundaries in the Wagiman examples.
Complex predicates crosslinguistically

b. nangay-nangay-ma nga-yu nung // “mamak”
  wave-REDUP-ASP 1SG-BE.PRS 3SG.OBL // “goodbye”
  nga-ha nung
  1SG.SAY.PRES 3SG.OBL

‘I am waving to him; “goodbye” I say to him.’

(Wagiman, Wilson 1999:114)

It seems that these coverbs may have an optional oblique argument. An
optional argument is also part of the argument structure of coverbs of emo-
tion, for which Wilson (1999) notes that many may take a dative argument
expressing the cause of emotion.

Finally, the class of coverbs of impact and violence is a class of transitive
coverbs and, as Wilson (1999) notes, they usually combine with transitive
inflecting verbs to form transitive complex predicates. In constructions with
intransitive inflecting verbs, mainly with the inflecting verb -ya- ‘go’, the
complex predicate still takes two arguments. However, Wilson (1999) argues
that the status of the object argument is not completely clear, as combina-
tions with intransitive inflecting verbs seem to be instances of pseudo-noun
incorporation. For example, in (2.44) both instances involve the coverb bik
‘clean’, but (2.44a) involves the transitive inflecting verb -bu ‘hit’ while
(2.44b) involves the intransitive inflecting verb -ya- ‘go’. Wilson (1999) ar-
gues that in (2.44b), the object lahan ‘camp’ is non-specific, so that an even
better translation might be ‘I went camp-cleaning’.

(2.44) a. lahan bik-bik ma-bu
  camp clean-RDP 1SG-HIT.FUT

  ‘I will clean up camp.’

  (Wagiman, Wilson 1999:121)

b. ngagun bik-ga nga-ya-nggi lahan yimbama
  1SG clean-ASP 1SG-GO-PST camp always

  ‘me, I went around cleaning up the place all day.’

  (Wagiman, Wilson 1999:122)

for modeling the combinatory possibilities of inflecting verbs and coverbs.
However, he replaces the Pred value of verbs in the functional structure with
the LCS of a verb. In this way, he differs from Butt’s (1995) account that
 treats the Lexical Conceptual Structure as an elaborate argument structure
which is then linked to the functional structure. In contrast, Wilson (1999)
uses LCSS as part of the representation in the functional structure (c.f. Alsina
1996). He adopts this idea from Andrews & Manning (1999).\(^6\)

\(^6\)Andrews & Manning (1999) change the function of the f-structure considerably. In
addition to replacing PREDs with LCSS, Andrews & Manning (1999) also introduce TERMS
to change the modeling of the dependency of heads and arguments. Wilson (1999) takes on
In addition to this change, Wilson (1999) also proposes different rules for the combination of the LCSs in complex predicate formation. This is necessary because the combinatory possibilities of complex predicates differ for Urdu and Wagiman. In Urdu complex predicates, the LCS of the light verb explicitly specifies where the LCS of the verb should be merged into. This is not possible in Wagiman as inflecting verbs and coverbs may combine in various ways. Wilson’s (1999) idea for Wagiman is that coverbs and inflecting verbs can form a complex predicate whenever their LCSs share enough structure to be merged. This is probably best explained by an example.

As has been discussed above and shown in (2.42a,d), state coverbs can combine with stative inflecting verbs while coverbs of change of state cannot. Wilson (1999) explains this with the fact that one can merge the LCSs of *guk* ‘sleep’ into the LCS of *-yu-* ‘BE’, but not the LCS of *bort* ‘die’ into the LCS of *-yu-* ‘BE’. This contrast can be observed in (2.45a) and (2.45b). The LCS of *guk* ‘sleep’ can combine with the LCS of *-yu-* ‘BE’, because both are states and the more detailed information for place in the LCS of *guk* ‘sleep’ can fill the underspecified place in the LCS of *-yu-* ‘BE’. The LCS of the complex predicate *guk-yu-*, thus, is the same as the LCS of *guk* ‘sleep’ in (2.45a).\(^7\)

\begin{enumerate}
\item *guk* ‘sleep’
[\text{State BE}_{\text{Ident}} ([\text{Thing } A \triangleright \text{Place AT}_{\text{Ident}} ( [\text{Property asleep}] ) ])]
\item *bort* ‘die’
[\text{Event BECOME} ([\text{State BE}_{\text{Ident}} ([\text{Thing } A \triangleright \text{Place AT}_{\text{Ident}} ( [\text{Property dead}] ) ])] )]
\item *-yu-* ‘BE’
[\text{State BE} ([\text{Thing } A \triangleright \text{Place } - ])]
\end{enumerate}

The change of state coverb *bort* ‘die’, however, cannot combine with the inflecting verb *-yu-* ‘BE’ because the two LCSs cannot be merged in an appropriate way. The inflecting verb determines the general shape of the LCSs of the complex predicates, which means that only the LCS of the coverb can be merged into the LCS of the inflecting verb, not vice versa. The combination *bort-yu-* ‘die-BE’ is ungrammatical because the LCS of the coverb cannot be merged into the LCS of the inflecting verb *-yu-* ‘BE’.

In contrast, coverbs of state and coverbs of change of state can combine with inchoative inflecting verbs, as was shown in (2.42b,c). Both state coverbs such as *dewh* ‘be white’ as well as coverbs of change of state like *wal* Andrews & Manning’s (1999) idea of including the LCSs into the f-structure, but keeps to the simpler mapping mechanisms between the arguments in the LCSs and the grammatical functions in the f-structure used by Butt (1995).

\(^7\)Wilson (1999) does not use Jackendoff’s (1990) Action Tier. However, for linking he marks the arguments that are required by the verb with a subscripted A.
'grow' can combine with the inchoative inflecting verb -ya(ma) - 'BECOME' to form an inchoative complex predicate. The LCSs of the two coverbs are provided in (2.46a,b), the LCS of the inflecting verb in (2.46c). Both LCSs of the coverbs can be merged into the LCS of the inflecting verb, just at different places, as can be seen in the combined LCS in (2.47). The LCS of the coverb of state dewh 'be white' can be merged into the LCS of the inflecting verb -ya(ma) - 'BECOME' because the LCS of -ya(ma) - 'BECOME' has an embedded state in its LCS with which the LCS of dewh 'be white' can merge. The LCSs of the coverb wal 'grow' and the inflecting verb -ya(ma) - 'BECOME' share their complete structure and can so be merged on the top level.

(2.46) a. dewh 'be white'

\[
\text{[Event BECOME ([State BE Ident ([Thing ]A :: [Place AT Ident ([Property white])])])]
\]

b. wal 'grow'

\[
\text{[Event BECOME ([State BE Ident ([Thing ]A :: [Place AT Ident ([Property big])])])]
\]

c. -ya(ma) - 'BECOME'

\[
\text{[Event BECOME ([State BE ([Thing ]A :: [Place —])])])]
\]

(2.47) a. dewh -ya(ma) - 'become white'

\[
\text{[Event BECOME ([State BE Ident ([Thing ]A :: [Place AT Ident ([Property white])])])]
\]

b. wal -ya(ma) - 'grow'

\[
\text{[Event BECOME ([State BE Ident ([Thing ]A :: [Place AT Ident ([Property big])])])]
\]

Wilson’s (1999) account of Wagiman complex predicates is very intriguing as the proposed rules for the combination of inflecting verb and coverb are very simple: they form a grammatical complex predicate if they share the structure of their LCSs such that the LCS of the coverb can be merged into the LCS of the inflecting verb. However, some combinations of inflecting verb and coverb exist that do not fit this neat pattern easily.

Firstly, Wilson (1999) has to account for the data in (2.44b) in which the inflecting verb -ya - 'GO' combines with the transitive coverb bik 'clean'. A similar example is provided in (2.48). In both examples, it seems that the inflecting verb contributes aspectual information rather than providing the meaning of motion along a path.

(2.48) gartgart-da ga-ya yimbama

\[
\text{laugh-ASP 3SG-GO.PRS always}
\]

‘he is always laughing’  

(Wagiman, Wilson 1999:157)
Wilson (1999) proposes to model this use of the inflecting verb -ya- ‘GO’ with the LCS in (2.49). The XFUNC is a placeholder entry that specifies where the LCS of the coverb should be merged into. This is very similar to Butt’s (1995) approach specifying transparent events.

(2.49)  
-ya- ‘GO’  
\[ \text{Event GO ([\text{Thing }] \alpha)} \text{ [ WITH ([XFUNC([\alpha])])]} \]

Wilson (1999) thus needs a second LCS for the inflecting verb -ya- ‘GO’ and an additional rule to account for the data in (2.48). Similarly, Wilson (2006) proposes further extensions to the set of rules for Wagiman complex predicates to model the combinatory possibilities of a range of combinations that were excluded in Wilson’s (1999) account. For example, Wilson (2006) states that combinations of activity coverbs and stative inflecting verbs are grammatical if the coverb does not specify a path. He discusses the example in (2.50) in which the activity dabulp ‘smoke’ combines with the inflecting verb -yu- ‘BE’.

(2.50)  
dabulp-pa ga-yu bakka  
smoke-ASP 3SG.PRS-BE tobacco  
‘He’s smoking tobacco.’ (Wilson 2006:30)

The additions to the formalism proposed by Wilson (2006) are still minor, but the system as such is not as neat and clean as proposed by Wilson (1999). I will demonstrate in Chapter 4 that it is very difficult to extend the approach to the Murrinh-Patha data, as classifier stems in Murrinh-Patha can exhibit a wide range of different meanings. An approach which only builds on the compatibility of the LCSs of the classifier and lexical stem in Murrinh-Patha would involve many different rules of how the LCSs could combine and would thus lose explanatory power.

2.4.2 Baker & Harvey (2010): Merger versus coindexation

Baker & Harvey (2010) use insights from Butt (1995) and Wilson (1999) to define a crosslinguistic typology of complex predicates. In their view, complex predicate formation can be grouped into two distinct subgroups, merger structures and coindexation structures and they use LCSs to model the difference.

In merger constructions, in their view, the LCSs of the elements building the complex predicate merge when they share a certain degree of structure. This comprises Butt’s (1995) event fusion analysis as well as the standard

---

8Wilson (1999) uses the GO function here as motion without a path. Why he does not use MOVE, which was used by Jackendoff (1990) as function for motion without path, is not clear to me.
formalism applied by Wilson (1999) to Wagiman complex predicates. In contrast, coindexation constructions restrict combinatory possibilities by posing a constraint on the coindexation of some of the arguments of the elements building the complex predicate. As such, Butt’s (1995) argument fusion analysis as well as Wilson’s (1999) analysis involving xfunc belong to this class of constructions.

Following work by Dowty (1979) and Levin & Rappaport Hovav (2005), among others, they propose that monomorphemic predicates are constrained in that their lcs can only contain the major predicate functions (cause, become, move, be) once and that they can only appear in the order as specified in (2.51).

(2.51) \text{CAUSE} > \text{BECOME} > \text{BE} \\
\text{CAUSE} > \text{MOVE}

Baker & Harvey (2010) claim that merger constructions group with these kinds of monomorphemic predicates while coindexation constructions do not need to follow these constraints.

As prototypical examples of merger constructions they suggest Australian coverb constructions and use examples from Marra (Heath 1981), Wagiman (Wilson 1999) and Jaminjung (Schultze-Berndt 2000) for this purpose. In contrast, coindexation constructions are most commonly found in serial verb constructions. Nevertheless, exceptions can be found as well. They use Ngan’gityemerri as example in which coverb constructions of associated posture and motion can be found as in (2.52).

(2.52) a. \text{ngirim-fifi} \\
1SG.SIT.PRS-smoke \\
‘I am sitting smoking.’ (Ngan’gityemerri, Reid 2000:338)

b. \text{ngirribem-fifi} \\
1SG.STAND.PRS-smoke \\
‘I am standing smoking.’ (Ngan’gityemerri, Reid 2000:338)

Baker & Harvey (2010) claim that these constructions cannot receive a merger analysis as they do not follow the patterns for monomorphemic predicates given in (2.51). Instead, a coindexation account is needed for these constructions in which the LCS modeling the posture and the LCS modeling the smoking are coindexed.

As Baker & Harvey (2010) point out, similar constructions can also be found in Murrinh-Patha and other Western and Northern neighbors of Ngan’gityemerri. They claim that these languages are also exceptional in a range of other properties of the coverb constructions, such as the inventory of inflecting verbs, as was discussed above, and the linear ordering of coverbs.
and inflecting verbs and conclude that these features may be dependent on one another.

For Baker & Harvey (2010), the associated posture and motion construction in Ngan’gityemerri and related languages constitutes an exception in the Australian coverb constructions. They claim that most other coverb constructions are instances of merger constructions. However, as was seen in the previous section in (2.48) and (2.50), Wagiman has a very similar construction. Similarly, Urdu seems to have merger and coindexation structures for its different light verb constructions as well. It is therefore questionable whether it makes sense to divide the two constructions across merger and coindexation languages, as Baker & Harvey (2010) propose.

2.4.3 Jaminjung complex predicates (Schultze-Berndt 2000)

Schultze-Berndt (2000) discusses complex predicates in the two closely related Northern Australian languages Jaminjung and Ngaliwurru which are spoken in the Victoria River district. In these two languages, as in Wagiman, the complex predicate consists of two morphologically distinct words. According to Schultze-Berndt (2000), there are approximately 35 inflecting verbs in these languages of which 26 are used frequently in complex predicate formations.

In the very detailed discussion of these complex predicates, Schultze-Berndt (2000) takes on a functionalist perspective that assumes that “language is used to convey meaning” (Schultze-Berndt 2000:23). She is interested in the question of openly expressed category systems, such as the categorization of coverbs according to the inflecting verbs they may combine with, and what such openly expressed category systems may tell us about the way people categorize the world. In her view, the inflecting verbs categorize the events encoded by the complex predicate, i.e., by the combination of inflecting verbs and coverbs.

Her treatment is highly influenced by construction-based approaches to languages. She assumes that constructions are meaningful, i.e., she assumes that the meaning of a complex expression cannot be predicated solely by the meaning of its parts, but that the construction itself contributes meaning to the complex expression. As such, her approach follows work such as Goldberg’s (1997) treatment of the English ‘way’-construction as complex predicate. The construction-based approach also influences the view of the lexicon, the division between morphology and syntax and the view on argument structure that Schultze-Berndt (2000) adopts.

When discussing the different combinatory possibilities, she argues for constraints on three different levels. Firstly, she considers the argument structure of complex predicates. The main idea is that of argument sharing similar to the approaches taken by Butt (1995), Wilson (1999) etc. That is, she assumes that both inflecting verbs and coverbs have argument slots and
in complex predicate formation, some or all of these argument slots may be unified.

For this purpose, she assumes that inflecting verbs and coverbs can be intransitive, transitive or ditransitive, i.e., that inflecting verbs and coverbs can have one, two or three core arguments or central participants, in a sense that she carefully defines. As Schultze-Berndt (2000) states, intransitive inflecting verbs can combine with intransitive coverbs to form intransitive complex predicates such as in (2.53a). A subset of the intransitive inflecting verbs, namely the inflecting verbs -yu ‘BE’ and -ijga ‘GO’ can also be used with transitive coverbs, conveying aspectual meaning ((2.53b)).

(2.53)  
\[ \text{walninginy ga-ngga \hspace{1em} buyi} \]
\[ \text{walk \hspace{1em} 3SG-GO.PRS \hspace{1em} keep.going} \]
\[ \text{‘he keeps on walking’ (Jaminjung, Schultze-Berndt 2000:194)} \]
\[ \text{thawaya=biya \hspace{1em} burr-inyi \hspace{1em} buliki} \]
\[ \text{eating=NOW \hspace{1em} 3PL-BE.IMPF \hspace{1em} cow} \]
\[ \text{‘they were eating cattle’} \]
\[ \text{(Jaminjung, Schultze-Berndt 2000:195)} \]

Transitive inflecting verbs, according to Schultze-Berndt (2000), can combine with both intransitive, transitive or ditransitive coverbs. If a transitive inflecting verb is combined with an intransitive coverb, the resulting complex predicate is normally transitive, as in (2.54).

(2.54)  
\[ \text{gurdiy \hspace{1em} gan-ngayi-m=mindag, \hspace{1em} mung} \]
\[ \text{stand \hspace{1em} 3SG>1-SEE-PRS=1DU.INCL.OBL \hspace{1em} watch} \]
\[ \text{‘he looks at us, standing’ (Jaminjung, Schultze-Berndt 2000:196)} \]

For these cases Schultze-Berndt (2000) points out that both the actor or the undergoer specified by the inflecting verb may be identified with the argument of the coverb. In (2.54), for example, it is the actor which is identified with the argument of the coverb mung ‘watch’.

In some cases, however, if a transitive inflecting verb combines with an intransitive coverb, the resulting complex predicate is intransitive. This can be seen in (2.55) in which the transitive inflecting verb hit ‘emerge’ to form an intransitive complex predicate.

(2.55)  
\[ \text{Ngayin=malang \hspace{1em} bul \hspace{1em} gani-ma \hspace{1em} bunyaag.} \]
\[ \text{meat.animal=given \hspace{1em} emerge \hspace{1em} 3SG>3SG-HIT.PST \hspace{1em} 3DU.OBL} \]
\[ \text{‘the animal came out to/for the two.’} \]
\[ \text{(Jaminjung, Schultze-Berndt 2000:181)} \]

---

9In fact, Schultze-Berndt (2000) uses the terms monovalent, bivalent and trivalent in her analysis. As the distinction between valency and transitivity is not important in this work, the terms intransitive, transitive and ditransitive are used.
Although the complex predicate as a whole is intransitive, the inflecting verb is marked with the transitive bound pronominal \( gani \). There is thus a non-correspondence between the morphology which requires a prefix and the semantics as the dummy prefix does not correspond to any participant. For cases like this, Schultze-Berndt (2000) assumes that the undergoer prefix is a dummy prefix similar to expletive subjects in expressions like \( \text{it is raining} \). As Bowern (2004) points out, transitive inflecting verbs are rarely part of intransitive complex predicates in Australian coverb constructions, but this pattern is attested in Bardi (Bowern 2004) and Murrinh-Patha (see Chapter 3.4) as well.

In cases in which transitive inflecting verbs combine with transitive coverbs, both arguments are shared. An example is provided in (2.56a). If a ditransitive coverb is involved, the resulting complex predicate is ditransitive, too ((2.56b)).

(2.56) a. \( \text{gugu burlug nga-minda-ny,} \)
\quad \text{water drink 1SG>3SG-EAT-PST}
\quad \text{‘I drank water’ (Jaminjung, Schultze-Berndt 2000:202)}

b. \( \text{mulurru-ni gagawuli yurrg gan-karra-ny} \)
\quad \text{old.woman-ERG long.yam show 3SG>1SG-PUT-PST}
\quad \text{Gilwi-loc}
\quad \text{‘the woman showed me yam in Gilwi’}
\quad \text{(Jaminjung, Schultze-Berndt 2000:162)}

For the discussion of the semantics of Jaminjung complex predicates, Schultze-Berndt (2000) uses a “semi-standardized English” (Schultze-Berndt 2000:32) as metalanguage because she wants to avoid to be constrained by formal metalanguages such as LCSs proposed by Jackendoff (1990).

In addition to the semantic principles, Schultze-Berndt (2000) also discusses pragmatic principles that constrain complex predicate formation. In her view, the difference between semantic meaning and pragmatic interpretation is one of degree in the way that semantic knowledge can be considered a subset of encyclopedic knowledge. Both levels are important for restraining the combinatory possibilities in complex predicate formation. One example that Schultze-Berndt (2000) discusses for this distinction is the combination of coverbs with inflecting verbs of contact and force. In this set of verbs, more general and more specific verb meanings can be found. The inflecting verb \(-ma\) ‘hit’ is used for events in which impact or force plays a role generally. Additionally, it is used to encode impact with a flat hand. The inflecting verbs \(-ina/(ngga)\) ‘CHOP’, \(-inama\) ‘KICK/STEP’ and \(-ijja/-yaluga\) ‘POKE’ specify the type of the instrument with which force is applied. An example is given in (2.57).
Schultze-Berndt (2000) assumes that the verb ma ‘HIT’ has the semantic meaning ‘x makes an impact on y, x affects y’. Pragmatic principles then explain how -ma ‘HIT’ contrasts with the more specific verbs -ina(ngga) ‘CHOP’, -inama ‘KICK/STEP’ and -ijja / -yaluga ‘POKE’. The pragmatic principle of “Quantity” that she introduces and that corresponds to Grice’s First Maxim of Quantity ensures that -ma ‘HIT’ is only used if none of the more specific verbs can be used to characterize the event more specifically. In this way, -ma ‘HIT’ is only used if the specific impact is not known or not important or if the impact cannot be described by one of the more specific verbs. In this case, the complex predicate formed with -ma ‘HIT’ can receive a default interpretation of impact with the flat hand.

This interplay of semantic meaning and pragmatic principles allows Schultze-Berndt (2000) to maintain monosemy for most lexical entries. For example, she does not have to define different meanings for the verb -ma ‘HIT’ to encode force and contact with a flat hand such as in (2.57a) or to encode amorphous natural forces such as rain or wind which also use the verb -ma ‘HIT’. The general meaning given to -ma ‘HIT’ accounts for both of these (and other) uses.

However, Schultze-Berndt (2000) does not pursue a strict monosemic view in that she gives separate lexical entries for uses of some of the inflecting verbs which exhibit a great range of variety in their combinatorial possibilities. One example is again the inflecting verb -ma ‘HIT’ which besides its impact/force meaning can also be used to express complete affectedness such as in (2.58).

(2.57) a. *gani-ma-m jurrung-ni*
   3SG>3SG-HIT-PRS lower.arm-ERG/INSTR
   ‘he slaps him (he hits him with the flat hand)’

   b. *gana-m jurrung-ni*
   3SG>3SG.CHOP-PRS lower.arm-ERG/INSTR
   ‘he hits him with the fist’

(2.58) walig *gani-ma-m gurrurrij*
   around 3SG>3SG-HIT-PRS car
   ‘he walks around the car’ (Jaminjung, Schultze-Berndt 2000:314)

To sum up, in Schultze-Berndt’s (2000) approach to complex predicates in Jaminjung, argument structure, lexical-semantic meaning and pragmatic principles restrict the combinatorial possibilities of coverbs and inflecting verbs. It seems that in Jaminjung, the argument structure functions as a first filter, while semantics and pragmatics then further restrict the possibilities. However, in some combinations, the argument structure does not...
act as a filter, e.g. when the inflecting verb contributes aspectual information as in (2.53b) or in cases like (2.55) in which a dummy prefix is involved. As will be seen in Chapter 3, Murrinh-Patha has even more exceptions like these so that applying the argument structure as a first filter is not very helpful for Murrinh-Patha.

2.4.4 Bardi complex predicates and Bowern’s (2004) analysis

Bardi is a Nyulnyulan language spoken in the Northern Territory of Australia. Bowern (2004, 2008b) is concerned with synchronic as well as with diachronic aspects of complex predicates in Bardi and Nyulnyulan languages more generally. She calls the uninflecting element a ‘preverb’, which combines with an inflecting verb to form a complex predicate. According to Bowern (2004), the preverb class is an open word class in Bardi in which loanwords can be borrowed. The class of inflecting verbs is closed, with approximately 230 verb roots. Of these verb roots, 12 are very commonly used as light verbs and 27 more can be used with a very restricted set of preverbs (Bowern 2004:145). Although Bardi has this larger number of inflecting verbs in comparison to e.g., Jaminjung or Ngan'gityemerri, complex predicates are still widely used, approximately three times as often as simple verbs (Bowern 2004:247). In contrast to Wagiman, the preverb cannot be used without an inflecting verb as a verbal predicate in Bardi.

For the analysis, Bowern (2004) follows the assumption that complex predicates are formed by transferring the argument structure and θ-roles from the preverb to the light verb. She therefore differs from e.g. the treatment of Wagiman complex predicates by Wilson (1999) who proposes that arguments are merged. For details of the analysis, she follows Hale & Keyser (2002) and Samek-Lodovici (2003). She also proposes a division of the Bardi complex predicates into four different subtypes to explain how the argument transfer works in detail for these subtypes.

With the first subtype, she covers cases in which the meaning of the preverb is very similar to the meaning of the inflecting verb, i.e., cases in which the meaning of the preverb is a subset of meaning of the light verb and so reinforces the meaning of the light verb. Examples of this type of complex predicate are verbs of speech or verbs of motion. For example, the light verb -nganka- ‘speak’ can combine with the preverb balygarr ‘swearing’ to form a complex predicate meaning ‘swear’. Similarly, the light verb -galala- ‘move’ combines with the preverb joodarrarr ‘in the direction of tidal motion’ to encode the meaning ‘go with the tide’. For this type of complex predicate, Bowern (2004) proposes that the arguments from the preverb are transferred to the light verb, but that this transfer is vacuous as the arguments of the preverb and the light verb are the same anyway.

Bowern (2004) also includes into this subgroup complex predicates which are formed with adverbial preverbs such as the complex predicate
ngaada -joogooloo- ‘break in half’ formed with the light verb -joogooloo- ‘BREAK’ and the preverb ngaada ‘short’. She states that this type of preverb does not seem to transfer θ-roles to the verb and that consequently, no argument transfer takes place either. She assumes a kind of ‘instrument incorporation’ for these constructions.

As a second type of complex predicate found in Bardi, Bowern (2004) considers a type of pseudo-incorporation in which the object or the instrument is pseudo-incorporated. An example is provided by the combination of the light verb -gonboo- ‘SEND’ and the preverb ngaanka ‘word, language’ to denote the meaning ‘send a message’.

The third type of complex predicate found in Bardi, Bowern (2004) argues, is that of classification, i.e., in which the light verb “functions as an event classifier and/or a licensor of various arguments” (Bowern 2004:303). She states that this is the most common type of complex predicate in Bardi. For this type of complex predicate, she follows previous work by Schultze-Berndt (2000) for Jaminjung, McGregor (2002) for Nyulnyulan languages and Nicolas (2000) for Bardi who propose that complex predicates in the respective languages can be classified according to valency, aspect and trajectory. For example, the light verb “-ar- ‘SPEAR LICE’ is used with actions which occur around a point and have a defined trajectory” (Bowern 2004:307) such as in dirray -ar- ‘rotate something’, jiin -ar- ‘point at something’ or jibiny -ar- ‘thrust a spear’.

While Bowern (2004) agrees with the main underlying ideas of Nicolas (2000), i.e., that light verbs classify events in Bardi, she disagrees with many of the statements concerning the possible combinations of preverbs and light verbs. She also disagrees with McGregor’s (2002) view that an analysis of the combination as classifier constructions excludes an analysis as complex predicate constructions. She instead shows how the combinations can be described in terms of a classifier analysis and be given a complex predicate analysis by discussing the semantics of the light verbs and the possible combinations with preverbs in detail. Finally, the fourth type of complex predicates is a residue category which Bowern (2004) assumes to be idioms in which the meaning composition is not clear.

Bowern (2004) further points out that in Bardi, apparently transitive inflecting verbs can be used in intransitive complex predicates. An example is provided in (2.59) in which the light verb -nya- ‘CATCH’ is marked with the transitive marker n but nevertheless forms an intransitive complex predicate. Bowern (2004) states that such examples may be problematic because the valency requirement of the light verb is not fulfilled in these cases, e.g., it violates the θ-criterion or the completeness requirements in LFG terms.

(2.59) Majoonggoooloo roowil i-n-nya-gal barda.
young girl walk 3-TR-CATCH-REC.PST away
‘The young girl walked off.’ (Bardi, Bowern 2004:304)
To sum up, the uniqueness of Bowern’s (2004) analysis lies in the very careful distinguishing of subtypes of complex predicates in Bardi and the different analyses she proposes for the different subtypes.

2.4.5 The Ngan’gityemerri system (Reid 2000)

Ngan’gityemerri was shown to be related to Murrinh-Patha by Green (2003), and a detailed look at the Ngan’gityemerri data should therefore also help with the complexity of the Murrinh-Patha data. Ngan’gityemerri has received comparably detailed linguistic description, mainly thanks to Nicholas Reid (e.g., Reid 1990, 2000, 2002, Reid & McTaggart 2008, Reid 2011). In a diachronic study, Reid (2003) could make use of earlier descriptions of the language by Tryon (1974) and Hoddinott & Kofod (1988) from the late 1960s and early 1970s and from fieldnotes collected by Gerhardt Laves in the 1930s. Reid (2003) shows how the differences between Ngan’gityemerri and the other Daly river languages, i.e., the reverse ordering of coverb and inflecting verb and the incorporation of both parts into one morphological word, evolved.

For the purpose of discussing the combinatory possibilities of classifier and lexical stems in Ngan’gityemerri, Reid (2000) divides the classifier stems and the lexical stems into different classes of transitivity. For classifier stems, Reid (2000) proposes a three-way distinction into intransitive, transitive and reflexive/reciprocal classifier stems. He infers the transitivity of the classifier stems based on the transitivity they display when used as the sole verbal predicate or the combinations they most typically occur in, e.g., if a classifier stem occurs most typically in a transitive complex predicate, it is considered to be transitive. Table 2.1 provides an overview over the Ngan’gityemerri classifier stems. There are 31 classifier stems, of which 11 can form a verbal predicate without a lexical stem.

For lexical stems, Reid (2000) uses a three-way distinction as well. He differentiates between intransitive, low transitive and high transitive lexical stems. The difference between the two classes of transitive lexical stems is that low transitive lexical stems specify activities in which the object is usually non-individuated and non-affected. In contrast, the object is usually individuated and affected for high transitive lexical stems.

From the three-way distinctions for classifier and lexical stems, 9 different possible combinations can be formed, of which 8 are attested according to Reid (2000). Basically, Reid (2000) can predict whether the resulting complex predicate is intransitive, transitive or reflexive. Additionally, he can predict finer-grained lexical semantic distinctions such as whether the resulting complex predicate is a causative complex predicate, an anticausative

\[\text{In fact, Reid (2000) uses the terms monovalent and bivalent in his treatment. Similarly to the discussion about the Jaminjung data, these terms have been changed to intransitive and transitive to ensure compatibility with the rest of the thesis.}\]
Complex predicates crosslinguistically

<table>
<thead>
<tr>
<th>Intransitive</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SIT</td>
<td>carried out in a sitting posture</td>
</tr>
<tr>
<td>STAND</td>
<td>carried out in a standing posture</td>
</tr>
<tr>
<td>LIE</td>
<td>carried out in a lying posture</td>
</tr>
<tr>
<td>GO</td>
<td>carried out in motion</td>
</tr>
<tr>
<td>TRAVEL</td>
<td>carried out in motion</td>
</tr>
<tr>
<td>PERCH</td>
<td>carried out up off the ground</td>
</tr>
<tr>
<td>ARRIVE</td>
<td>involving arrival/emergence</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transitive</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SAY/DO</td>
<td>speech and unspecified doing</td>
</tr>
<tr>
<td></td>
<td>(do things, say things)</td>
</tr>
<tr>
<td>SEE</td>
<td>performed with the eyes</td>
</tr>
<tr>
<td></td>
<td>(look at, watch, keep an eye on  )</td>
</tr>
<tr>
<td>TAKE</td>
<td>taking/bringing things</td>
</tr>
<tr>
<td>POKE</td>
<td>using long thin things in point</td>
</tr>
<tr>
<td></td>
<td>contact (stab, prod)</td>
</tr>
<tr>
<td>SLASH</td>
<td>using hinged trajectory and edge-</td>
</tr>
<tr>
<td></td>
<td>on contact (sweep, slice)</td>
</tr>
<tr>
<td>HANDS</td>
<td>holding things within the grasp</td>
</tr>
<tr>
<td></td>
<td>of the hands (grab, hold, grip)</td>
</tr>
<tr>
<td>FEET</td>
<td>holding things down with the feet</td>
</tr>
<tr>
<td></td>
<td>(tread on, kick, walk on)</td>
</tr>
<tr>
<td>MOUTH</td>
<td>holding things within the mouth</td>
</tr>
<tr>
<td></td>
<td>(chew, suck, some speech verbs)</td>
</tr>
<tr>
<td>BASH</td>
<td>using vertical trajectory and</td>
</tr>
<tr>
<td></td>
<td>lumpy contact (thump, crash)</td>
</tr>
<tr>
<td>MOVE</td>
<td>moving things to a different</td>
</tr>
<tr>
<td></td>
<td>place (shift, throw, push)</td>
</tr>
<tr>
<td>HEAT</td>
<td>applying heat (burn, melt, warm,</td>
</tr>
<tr>
<td></td>
<td>light)</td>
</tr>
<tr>
<td>SUCK</td>
<td>ingesting things (eat, drink)</td>
</tr>
<tr>
<td>PULL</td>
<td>pulling things (pull, tow, lever</td>
</tr>
<tr>
<td></td>
<td>up)</td>
</tr>
<tr>
<td>SNATCH</td>
<td>acquiring things (get, pick up)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detransitive/reflexive</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HANDS.REFL</td>
<td>reflexive activity holding things</td>
</tr>
<tr>
<td></td>
<td>within the grasp of the hands</td>
</tr>
<tr>
<td>FEET.REFL</td>
<td>reflexive activity holding things</td>
</tr>
<tr>
<td></td>
<td>down with the feet</td>
</tr>
<tr>
<td>MOUTH.REFL</td>
<td>reflexive activity holding things</td>
</tr>
<tr>
<td></td>
<td>within the mouth</td>
</tr>
<tr>
<td>POKE.REFL</td>
<td>reflexive activity using long</td>
</tr>
<tr>
<td></td>
<td>thin things in point contact</td>
</tr>
<tr>
<td>MOVE.REFL.DYN</td>
<td>reflexive activity of moving</td>
</tr>
<tr>
<td></td>
<td>things to a different place</td>
</tr>
<tr>
<td></td>
<td>(dynamic)</td>
</tr>
<tr>
<td>MOVE.REFL.STAT</td>
<td>reflexive activity of moving</td>
</tr>
<tr>
<td></td>
<td>things to a different place</td>
</tr>
<tr>
<td></td>
<td>(stative)</td>
</tr>
<tr>
<td>BASH.REFL</td>
<td>reflexive activity using vertical</td>
</tr>
<tr>
<td></td>
<td>trajectory and lumpy contact</td>
</tr>
<tr>
<td>HEAT.REFL</td>
<td>reflexive activity by applying</td>
</tr>
<tr>
<td></td>
<td>heat</td>
</tr>
<tr>
<td>SEE.REFL</td>
<td>reflexive activity performed</td>
</tr>
<tr>
<td></td>
<td>with the eyes (look at your own</td>
</tr>
<tr>
<td></td>
<td>reflection)</td>
</tr>
<tr>
<td>SAY/DO</td>
<td>reflexive speech (talk to</td>
</tr>
<tr>
<td></td>
<td>yourself, mutter under breath)</td>
</tr>
</tbody>
</table>

Table 2.1: Ngan’gityemerri’s 31 classifier stems and their meaning. Above the dotted line are classifier stems that can occur without a lexical stem. The classifier stems beneath the dotted line can only occur in complex predicates. Adapted from Reid (2000).
complex predicate and whether the complex predicate displays high or low transitivity. An overview (adapted from Reid 2000) is presented in Table 2.2. In the following, these patterns are briefly discussed.

(2.60) is an example of an intransitive classifier stem with an intransitive lexical stem. In these cases, Reid (2000) claims, the subject can fulfill agentive or patient roles.

(2.60) *Yenim-purity*

3SGS.GO.NFUT-slip

‘He slipped.’ (Ngan’gityemerri, Reid 2000:343)

If an intransitive lexical stem combines with a transitive classifier stem, a causative effect evolves. In (2.61), the transitive classifier stem *move* combines with the intransitive lexical stem *du* ‘sleep’ to form a causative meaning ‘put to sleep.’

(2.61) *Ngirringirr* ngudum-birki-du.

sleep 1SGA.MOVE.PST-3DU.DO-sleep

‘I put them to sleep.’ (Ngan’gityemerri, Reid 2000:344)

Low transitive lexical stems form different patterns with intransitive and transitive classifier stems. The lexical stem *wutu* ‘pour’, for example, can combine with both the intransitive classifier stem *go* and the transitive *move*. The resulting complex predicate has roughly the same meaning, but the sentences are distinct in their focused material. Reid (2000) claims that combining a low transitive lexical stem with an intransitive classifier stem results in focus on the subject movement and activity as in (2.62a). As

---

<table>
<thead>
<tr>
<th>Classifier stem</th>
<th>Lexical stem</th>
<th>Complex predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>intransitive</td>
<td>intransitive</td>
<td>intransitive verb</td>
</tr>
<tr>
<td>low transitive</td>
<td>(in)transitive with S/A focus</td>
<td></td>
</tr>
<tr>
<td>high transitive</td>
<td>intransitive anticausative</td>
<td></td>
</tr>
<tr>
<td>transitive</td>
<td>intransitive</td>
<td>transitive causative</td>
</tr>
<tr>
<td>low transitive</td>
<td>low transitive with O focus</td>
<td></td>
</tr>
<tr>
<td>high transitive</td>
<td>high transitive</td>
<td></td>
</tr>
<tr>
<td>reflexive</td>
<td>intransitive, causative/reflexive</td>
<td></td>
</tr>
<tr>
<td>low transitive</td>
<td>(unattested)</td>
<td></td>
</tr>
<tr>
<td>high transitive</td>
<td>intransitive, reflexive</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.2: Argument structure of Ngan’gityemerri classifier plus lexical stem combinations (adapted from Reid 2000).
the third person singular direct object marker is zero in Ngan’gityemerri, it is not clear if the resulting complex predicate should be considered transitive or intransitive and whether an A or S subject is involved. In contrast, (2.62b) involves the transitive classifier stem MOV in a low transitive complex predicate in which the focus is on how the object is manipulated.

(2.62)  

a. \textit{ngaganim(-\emptyset)-wut} (\textit{kur})  
\textit{1SGS/A.GO.NFUT(-3SG.DO)-pour} (water)  
\text{‘I poured the water out (focus on subject movement/activity).’}  
\textit{(Ngan’gityemerri, Reid 2000:345)}  

b. \textit{ngudupun-\emptyset-wut} (\textit{kur})  
\textit{1SGA.MOVE.NFUT-3SG.DO-pour} (water)  
\text{‘I poured the water out (focus on how object is manipulated).’}  
\textit{(Ngan’gityemerri, Reid 2000:345)}

High transitive lexical stems form a different pattern when they combine with transitive and intransitive classifier stems. If a high transitive lexical stem combines with a transitive classifier stem such as in (2.63a), a high transitive complex predicate results. If a high transitive lexical stem combines with an intransitive classifier stem such as in (2.63b), the argument structure of the lexical stem is reduced and an intransitive complex predicate is formed. Reid (2000) refers to this complex predicate formation as anticausative formation.

(2.63)  

a. \textit{ngerim-\emptyset-tum}  
\textit{1SGA.HANDS.NFUT-bury}  
\text{‘I sank it.’}  
\textit{(Ngan’gityemerri, Reid 2000:347)}  

b. \textit{ngirim-tum}  
\textit{1SGS.SIT.NFUT-bury}  
\text{‘I’m sinking.’}  
\textit{(Ngan’gityemerri, Reid 2000:347)}

Reflexive classifier stems can either combine with intransitive or high transitive lexical stems. In (2.64), the contrast between a high transitive lexical stem with a transitive classifier stem ((2.64a)) and with a reflexive classifier stem ((2.64b)) can be observed.

(2.64)  

a. \textit{ngeriny-\emptyset-syirr}  
\textit{1SGA.HANDS.PST-3SG.DO-scratch}  
\text{‘I scratched her.’}  
\textit{(Ngan’gityemerri, Reid 2000:348)}  

b. \textit{ngemeny-syirr}  
\textit{1SGS.HANDS.REFL.PST-scratch}  
\text{‘I scratched myself.’}  
\textit{(Ngan’gityemerri, Reid 2000:348)}
(2.65) shows the three different possible combinations with intransitive lexical stems. In (2.65a), the intransitive lexical stem *wurr* ‘enter’ combines with an intransitive classifier stem to form an intransitive complex predicate. If the same lexical stem is combined with a transitive classifier stem such as in (2.65b), a causative is formed. Finally, the lexical stem can also form the reflexive of the causative with a reflexive classifier stem as in (2.65c).

(2.65)  

a. *ngaganiny-wurr*  
\[ \text{1SGS.GO.PST}-\text{enter} \]  
‘I entered.’ (Ngan’gityemerri, Reid 2000:349)

b. *Ngudum-∅-wurr*  
\[ \text{1SGA.MOVE.PST-3SG.DO}-\text{enter} \text{ leg-ASSOCIATIVE} \]  
‘I dressed him. (I put him into trousers; causative)’  
(Ngan’gityemerri, Reid 2000:349)

c. *Ngudeny-wurr*  
\[ \text{1SGS.MOVE.REFL.DYN.PST}-\text{enter} \text{ leg-ASSOCIATIVE} \]  
‘I got dressed (I put myself into trousers; causative and reflexive)’  
(Ngan’gityemerri, Reid 2000:349)

The system that Reid (2000) proposes for Ngan’gityemerri complex predicate formation is very clear. It is a description of the valency alternations found in this language similar to the well studied argument alternations of verbs in other languages, for example of English verbs of *hitting* and *breaking* (e.g. Levin & Rappaport Hovav 2005, Fillmore 1970). In contrast to the English verbs, which do not change their surface form when argument alternation takes place, the verbs in Ngan’gityemerri use different classifier stems for this purpose.

A shortcoming of his approach is, however, that no independent diagnostics to distinguish between high and low transitivity lexical stems are provided. The distinguishing factor is that low transitivity lexical stems in combination with intransitive classifier stems result in a transitive complex predicate with an unaffected object and that they cannot combine with reflexive classifier stems. In contrast, the high transitivity lexical stems, when combined with intransitive classifiers stems, result in a clearly intransitive, anticausative complex predicate and they can combine with reflexive classifier stems. Independent diagnostics would provide a greater motivation for the existence of such a division.

This becomes especially clear when considering that these two classes are probably not the only two classes of transitive lexical stems found in Ngan’gityemerri. A third class can be identified as transitive lexical stems which are able to combine with transitive and reflexive classifier stems but do not combine with intransitive classifier stems to form anticausative complex predicates. Not surprisingly, this class includes a range of lexical stems
which do not have a causative relation inbuilt. An example can be seen in (2.66) in which the classifier stems see and see.refl combine with the lexical stem bebi ‘glance’.

(2.66)  
\begin{itemize}
  \item a. Membirr dede  
  \begin{align*}
  & \text{child} \quad \text{country} \\
  & \text{winninggin-madi-bebi=wannim} \\
  & \text{3PL.A.SEE.NFUT-IBP-glance=3PLS.GO.NFUT} \\
  & \text{‘The kids are looking at the country laid out on front of them.’} \\
  & \text{(Nganjgityemerri, Reid & McTaggart 2008:19)}
  
  \item b. Nginyerrem-mi-bebi  
  \begin{align*}
  & \text{1SGS.SEE.REFL.NFUT-IBP-glance} \\
  & \text{‘I checked my face (in the mirror).’} \\
  & \text{(Nganjgityemerri, Reid & McTaggart 2008:19)}
  \end{align*}
\end{itemize}

This demonstrates that the system, as neat as it is for the purpose of sorting through the combinations on a broader scale, is too coarse-grained when looking at the details. Not all transitive lexical stems which do not form a kind of pseudo incorporation with an intransitive classifier stem are automatically of high transitivity and form anticausatives. Some transitive lexical stems simply only combine with transitive and reflexive classifier stems and do not combine with intransitive lexical stems.

Moreover, it needs to be pointed out that not all classifier stems can combine with all lexical stems from these defined classes. The semantics play an important role as well, as e.g., the lexical stem bebi ‘glance’ can only combine with the two classifier stems above. An evaluation of the possible combinations of lexical and classifier stems with Reid & McTaggart (2008) as a database reveals that of the 650 lexical stems in the database, a large majority is only listed with one classifier stem (468 of 650 lexical stems) and a further large part is only listed with two different classifier stems (109 of 650 lexical stems).

Reid (1990), and its published version Reid (2011), discusses the semantics of some of the classifier stems in more detail. For example, he points out that hands can only be used when it describes a grasping or holding action while it cannot be used to describe actions in which only a short contact is involved, such as hitting or clapping. He also discusses that the classifier stem feet can be used in complex predicates of motion as well as in complex predicates in which the feet have an instrumental use and contrasts with the classifier stem hands in these cases. This ambiguity of the classifier stem feet can be found in the Murrinh-Patha data as well.

As Murrinh-Patha is closely related to Nganjgityemerri, many of the patterns discussed in this section can be found in Murrinh-Patha as well. However, some important differences can also be detected. In Chapter 3,
the patterns of Murrinh-Patha classifier and lexical stem combinations are discussed and it is argued that in Murrinh-Patha, it seems to be the lexical semantic content that primarily accounts for the combinatory possibilities, and not the argument structure.

2.4.6 The verbal classification approach (McGregor 2002)

Many works on Australian bipartite verbal structures assume some kind of verbal classification more or less implicitly, e.g., Capell (1979), McGregor (1990), Reid (1990, 2011), Nicolas (2000), Schultze-Berndt (2000), Bowern (2004). But by far the most explicit treatment of bipartite verbal constructions as verbal classification constructions in Australian languages is presented by McGregor (2002). In his approach, the inflecting verbs classify coverbs into classes, similarly to nominal classification. That is, the coverbs form classes based on their compatibility with inflecting verbs. Three different factors may play a role in the classification, i.e., valency, Aktionsart and vectorial configuration.

As he points out, this classification differs from verb classes as proposed by e.g., Fillmore (1968) or Levin (1993) who divide verbs based on transitivity, felicity etc. While these classifications are based on the inherent properties of the verbs, i.e., the classification is covert, the verbal classification in McGregor’s (2002) sense is overt. He further distinguishes his type of verbal classification with verbal classification in Amerindianist linguistics in which the verb serves to classify the object.

What is important to the discourse of Australian bipartite verbal constructions is the distinction between the verbal classification approach proposed by McGregor (2002) and the event classification approach proposed by others, most prominently Schultze-Berndt (2000), but also Bowern (2004). As laid out above, in the verbal classification approach the inflecting verb serves to classify the coverbs. In contrast, in the event classification approach, the inflecting verb serves to classify the event described by the coverb and inflecting verb together. I agree with Barone-Nugent (2008) in that both approaches are not mutually exclusive, as the event classification approaches usually encompass the verbal classification in discussing event classification based on the classes of coverbs that are determined by the compatibility of the coverbs with inflecting verbs.

McGregor (2002) further argues that Australian bipartite constructions should not be considered complex predicates and that consequently, fusion or union analyses are not applicable. His main reason for the rejection of a complex predicate analysis lies in his reservations towards the notion of a ‘head’. In his view, the definition of a syntactic head is not clear and even if it were, the light verbs and coverbs in Australian bipartite verbal complexes could not be heads at the same time. This, then, excludes the construction from a complex predicate analysis.
However, if a verbal head is defined as an element which determines the argument structure and semantics of an event, both inflecting verbs and coverbs are indeed heads and form a complex predicate together as co-heads. Bowern (2004) and Barone-Nugent (2008) share the view that a classification approach is not mutually exclusive with a complex predicate analysis.

2.5 Conclusion

This chapter provided a crosslinguistic overview over complex predicate constructions, how they can be distinguished from other multi-verb constructions and how complex predicates are analyzed in the various linguistic frameworks. It was shown that complex predicates can differ considerably from language to language and that therefore, different analyses might be required. This was true for complex predicates crosslinguistically in general and Australian coverb constructions more specifically.

Australian coverb constructions, in spite of their similarities at first glance, exhibit considerable variety in their details. This has implications both for the terminology as well as for descriptions and analyses of these constructions. The Daly river languages, modulo the differences between the languages, constitute a recognizable subgroup in the behavior of their coverb constructions. Murrinh-Patha is closely related to Ngan’gityemerri, and the detailed look at Ngan’gityemerri can therefore help to understand the Murrinh-Patha patterns. In the following chapter, the different patterns found in Murrinh-Patha complex predicates are considered in detail.
Complex predicates crosslinguistically
Chapter 3

Murrinh-Patha complex predicate patterns

The previous chapter outlined the main properties of Australian coverb constructions and the dimensions on which they differ.¹ Murrinh-Patha classifier and lexical stem combinations can be classified along with these coverb constructions as a non-inflecting element from a larger part of speech category, the lexical stem, combines with an inflecting element from a smaller part of speech category, the classifier stem. An example is provided in (3.1) in which the lexical stem rirda ‘push’ combines with the classifier stem hands(8).

(3.1) \( \text{marntirda} \)
\( \text{mam-rirda} \)
\( 3\text{SG.S.HANDS(8).NFUT-push} \)
‘He pushed him (with his hands).’  
(Seiss & Nordlinger 2010)

However, Murrinh-Patha differs from most other Australian coverb constructions in the combinatory possibilities that it allows. These combinatory possibilities are discussed in detail in this chapter from an argument structure point of view. It will be shown that in Murrinh-Patha, transitive classifier stems can be part of intransitive complex predicates, such as in (3.2). Bowern (2004) states that this is uncommon in Australian languages, but that it does occur in Bardi.

(3.2) \( \text{man-dharrwitj} \)
\( 3\text{SG.HANDS(8).NFUT-glide} \)
‘It glided.’  
(Barone-Nugent 2008)

¹Some of the complex predicate patterns discussed in this chapter were also described in Seiss & Nordlinger (2010). However, this chapter is a more detailed description and the conclusions also differ.
Similarly, Murrinh-Patha intransitive classifier stems can also occur in transitive complex predicates such as in (3.3). This pattern has been reported for Ngangi’tymerrri by Reid (2000) and is considered rather exceptional for Australian languages by Baker & Harvey (2010).

(3.3) *thamul pirrim-nga-batbat*

  spear 3SGS.STAND(3).NFUT-1SG.BEN-throw(RDP)

  ‘He always throws the spear at me.’ (Street 1989)

Approaches such as Wilson (1999), Schultze-Berndt (2000) or Reid (2000) all share the assumption that the argument structure plays a very prominent role in the selection process of the combinatory possibilities. That is, they assume that the argument structure functions as a first filter to exclude ungrammatical combinations and that further lexical semantic concepts might then restrict the combinatory possibilities even further. Cases such as (3.2) and (3.3) in these approaches then have to be treated as exceptions to the general pattern and explained with some additional machinery.

This thesis proposes a different approach to Murrinh-Patha classifier and lexical stem combinations. Murrinh-Patha complex predicates seem to involve a scale from very productive and predictable classifier plus lexical stem combinations to very idiosyncratic combinations. An example of a productive complex predicate was provided in (3.1) in which it is obvious which meaning parts are contributed by the classifier stem and the lexical stem. This becomes even more obvious when (3.1) is compared to (3.4), which uses the same lexical stem *rirda* ‘push’ but the classifier stem FEET(7).

(3.4) *ngunu-rirda-nu*

  1SGS.FEET(7).FUT-push-FUT

  ‘I will push it with my foot.’ (Street 1989)

In (3.1) and (3.4), the classifier stems contribute the instrument, i.e., they specify whether the action is carried out by hand or foot. In contrast, in idiosyncratic combinations it is not clear why this specific classifier stem is used. An example is provided in (3.5) in which it is not clear what the classifier stem contributes (c.f. Nordlinger 2011)

(3.5) *mam-pun-mardaraki*

  3SGS.HANDS(8).NFUT-3PL.DO-disappoint

  ‘He disappointed them(pl).’ (Nordlinger 2011)

In between these two endpoints, however, are a range of semi-productive combinations which can be predicted if certain aspects of the lexical semantic content of the classifier and lexical stems are taken into account. An approach that focusses on the argument structure of complex predicates would have to treat these semi-productive combinations as exceptions...
along the lines with the really idiosyncratic combinations. As this is not
desirable, this thesis instead proposes that semantic concepts play a role in
the selectional restrictions of Murrinh-Patha classifier and lexical stem com-
binations. The argument structure operations are then derived from the
different patterns that arise from the semantically determined groups. This
chapter presents the Murrinh-Patha data that is relevant for the claim that
it is not the argument structure that plays the most prominent part in the
selection process of Murrinh-Patha classifier and lexical stem combinations.
The semantic analysis will be described in detail in Chapter 4.

This chapter thus gives an overview over the various patterns found in
Murrinh-Patha classifier and lexical stem combinations from an argument
structure perspective. Assuming an argument structure for Murrinh-Patha
classifier and lexical stems, however, is not straightforward, as only a sub-
set of classifier stems and no lexical stems can function as the sole verbal
predicate. Section 3.1 therefore discusses how the valency of the complex
predicate as a whole as well as of the individual classifier and lexical stems
may be determined. The section proposes a working definition of the va-
leny of classifier and lexical stems, i.e., that the valency should be deduced
from the patterns they usually form in complex predicate formation.

This definition is used in this chapter when a classifier or lexical stem is
referred to as transitive or intransitive. It should be kept in mind, though,
that this is a working definition that has its problems, as Section 3.1 also
points out. The formal analysis proposed in Chapter 4 will not hinge on
the working definition as given in this chapter. The working definition only
serves as a means to organize the various combinations into patterns that
allow for comparing them with other descriptions of Australian coverb con-
structions and for testing other analyses such as Wilson (1999), Reid (2000)
and Schultze-Berndt (2000) against the Murrinh-Patha data.

Based on the definition of valency for classifier and lexical stems, the
most common pattern found in Murrinh-Patha complex predicates is argu-
ment structure matching. However, other patterns also exist in Murrinh-
Patha. Section 3.2 presents cases in which the argument structures do not
match, but in which the argument structure of the complex predicate as
a whole is determined by the classifier stem. This comprises intransitive
lexical stems used in transitive complex predicates with a causative read-
ing as well as transitive lexical stems combined with intransitive classifier
stems to form anticausative complex predicates. For most approaches to
Australian coverb constructions, the first pattern is not difficult to account
for. The second pattern involving anticausative formation, however, might
be problematic.

If the valency of the complex predicate differs from the valency of the
classifier stem, most analyses have to treat the combinations as exceptions.
Sections 3.3 and 3.4 discuss these cases. Section 3.3 discusses intransi-
tive classifier stems used in transitive complex predicates while Section 3.4
is concerned with transitive classifier stems used in intransitive complex predicates. Section 3.5 presents the classifier stem FEET(7) as an example of an ambiguous classifier stem and finally, Section 3.6 discusses reflexive/reciprocal classifier stems which can be used in complex predicates with various argument structures, too.

Section 3.7 concludes the discussion about the argument structure of Murrinh-Patha complex predicates and summarizes the findings that are needed for Chapter 4 which presents the semantic analysis of Murrinh-Patha complex predicates.

3.1 Valency of classifier and lexical stems

While it is relatively straightforward to determine the argument structure of the complex predicate as a whole, establishing the valency of the individual classifier and lexical stems is not easily accomplished as they usually cannot function as the sole verbal predicate in Murrinh-Patha. However, dividing classifier and lexical stems into different valency classes can be helpful in describing the various patterns of classifier and lexical stem combinations in Murrinh-Patha. This makes it possible to compare the Murrinh-Patha patterns to patterns of other Australian coverb constructions and to test existing analyses against the Murrinh-Patha data. This section thus provides a working definition of the valency of Murrinh-Patha classifier and lexical stems which then helps to describe the classifier and lexical stems combination patterns in the remainder of the chapter.

As Nordlinger (2011) shows, the number of arguments of the complex predicate as a whole can be determined based on morphological and syntactic clues. Human direct objects are marked on the verbal complex in Murrinh-Patha, with the exception of third person singular direct objects. The presence of such a direct object marker is thus treated as evidence that the complex predicate as a whole is transitive. (3.6a) presents an example involving the second person direct object marker nhi. Similarly, unmarked NPs with a patient/theme role can also function as a direct object and the complex predicate can thus be considered transitive as well. For example, the noun phrase kura patha ‘drinking water’ is considered the object in (3.6b).

(3.6)    a. nakurl ba-nhi-ngkardu-nu
        later 1SG.SEE(13).FUT-2SG.DO-SEE-FUT
        ‘I’ll see you later.’  (Seiss & Nordlinger 2010)

        b. kura patha ba-gurduk-nu
        NC_water good 1SG.SEE(13).FUT-drink-FUT
        ‘I will drink water.’  (Street 1989)
However, as Nordlinger (2011) also points out, the fact that third person human and all non-human direct objects are not overtly marked in the verbal template can make it difficult to decide on the transitivity of the complex predicate. Some examples involving human third person singular direct objects are clearly transitive in that they involve a patient argument nonetheless. Examples are provided in (3.7). However, some cases might not be as clear. Nordlinger (2011) deals with such cases in detail and actually proposes a continuum of transitivity for Murrinh-Patha complex predicates. Some of these examples are discussed in Section 3.3.

(3.7)  

a. nungarntirda  
   nungam-rirda  
   3SGS.FEET(7).NFUT-push  
   ‘He kicked him.’ (Seiss & Nordlinger 2010)

b. marntirda  
   mam-rirda  
   3SGS.HANDS(8).NFUT-push  
   ‘He pushed him (with his hands).’ (Seiss & Nordlinger 2010)

It should be pointed out here that in addition to direct object markers, Murrinh-Patha also marks what Nordlinger (2011) calls ‘benefactives’ on the verbal complex. These ‘benefactive’ markers can be used to encode benefactive/recipient arguments as well as adjuncts (Nordlinger 2011). (3.8a) provides an example of a benefactive/recipient argument while (3.8b) provides an example of an adjunct. In the following, I will concentrate on the argument structure as involving subjects and direct objects, but it should be kept in mind that the pattern might be even more complex than discussed due to the presence of benefactive/recipient arguments.

(3.8)  

a. parram-na-mut  
   kardu  
   numi  
   3PLS.POKE(19).NFUT-3SG.BEN.M-give  
   NC\textsubscript{human} one  
   ‘They gave them to one person.’ (Nordlinger 2011:710)

b. nga-nhi-kum-nu  
   1SGS.TAKE(22).FUT-2SG.BEN-swim-FUT  
   ‘I’ll swim it across for you.’ (Nordlinger 2011:710)

Determining the valency of the individual classifier and lexical stems is considerably more difficult. For the classifier stems which can occur on their own, the valency that they display when occurring on their own is assumed to be the valency that they also contribute to a complex predicate.

For classifier stems which cannot function as a sole verbal predicate, the valency can only be determined by looking at the multitude of possible combinations of lexical and classifier stems and working out the semantic meaning of the different parts of the complex predicate. That is, if a classifier
stem occurs most often in transitive complex predicates, it is assumed that the classifier stem is transitive, too.

This of course bears the risk of being a circular argumentation. However, because these classifier stems cannot occur on their own, there is no other way to determine their valency. Reid (2000) uses the same diagnostics for Ngan’gityemerri classifier and lexical stem combinations. In contrast, Jaminjung and Wagiman classifier stems can function as the sole verbal predicate, which makes it easier to establish the valency, as has already been discussed in Chapter 2.

Table 3.1 provides an overview over the Murrinh-Patha classifier stems and their assumed valency. The valency was established by using Street (1989), Joe Blythe’s toolbox dictionary as well as fieldnotes by Rachel Nordlinger as a database. The glosses for the classifier stems are adapted from Blythe et al. (2007) and Nordlinger (2008, 2011, 2015).

Some of the classifier stems could not be given a gloss because they do not occur often enough in the database. Similarly, the valency of these classifier stems could not be established without doubt so far. Classifier stems which occur less than 10 times with different lexical stems have thus been marked by * in the table and the assumed valency for these lexical stems in only tentative.

The problem with establishing the valency of these classifier stems beyond doubt lies in the fact that, as will be discussed in Sections 3.3 and 3.4, transitive classifier stems do not exclusively occur in transitive complex predicates and intransitive classifier stems not exclusively in intransitive complex predicates. The assumed valency for the classifier stem is based on the valency of the complex predicates it most often occurs in. If the database only has a few entries for a classifier stem with different lexical stems, it cannot be established without doubt whether this is indeed the valency that it most often occurs in.

(3.9) presents examples of the classifier stem 25 which occurs only 10 times in the database. In most of these cases, it occurs in transitive complex predicates such as in (3.9a), which even involves the third person plural direct object marker pun and can therefore be considered unambiguously transitive. However, the database also contains two combinations of the classifier stem 25 which seem to form intransitive complex predicates, as in (3.9b).

(3.9)

a. ngingam-pun-lip
   1SGS.25.NFUT-3PLDO-squash
   ‘I squashed them(pl) together’ (Street 1989)

b. nanthi tina yingamkangaweth=wurran
   nanthi tina yingam-gangaweth=wurran
   NCres sun 3SGS.25.NFUT-set=3SGS.GO(6).NFUT
   ‘The sun is setting (half visible).’ (Street 1989)
<table>
<thead>
<tr>
<th>Classifier stem</th>
<th>Assumed valency</th>
</tr>
</thead>
<tbody>
<tr>
<td>sit(1)</td>
<td>intransitive</td>
</tr>
<tr>
<td>lie(2)</td>
<td>intransitive</td>
</tr>
<tr>
<td>stand(3)</td>
<td>intransitive</td>
</tr>
<tr>
<td>be(4)</td>
<td>intransitive</td>
</tr>
<tr>
<td>perch(5)</td>
<td>intransitive</td>
</tr>
<tr>
<td>go(6)</td>
<td>intransitive</td>
</tr>
<tr>
<td>feet(7)</td>
<td>intransitive &amp; transitive</td>
</tr>
<tr>
<td>hands(8)</td>
<td>transitive</td>
</tr>
<tr>
<td>snatch(9)</td>
<td>transitive</td>
</tr>
<tr>
<td>hands:rr(10)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>break(11)</td>
<td>transitive</td>
</tr>
<tr>
<td>12</td>
<td>intransitive*</td>
</tr>
<tr>
<td>see(13)</td>
<td>transitive</td>
</tr>
<tr>
<td>bash(14)</td>
<td>transitive</td>
</tr>
<tr>
<td>bash:rr(15)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>16</td>
<td>intransitive*</td>
</tr>
<tr>
<td>17</td>
<td>transitive</td>
</tr>
<tr>
<td>18</td>
<td>intransitive*</td>
</tr>
<tr>
<td>poke(19)</td>
<td>transitive</td>
</tr>
<tr>
<td>20</td>
<td>intransitive</td>
</tr>
<tr>
<td>poke:rr(21)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>take(22)</td>
<td>transitive</td>
</tr>
<tr>
<td>slash(23)</td>
<td>transitive</td>
</tr>
<tr>
<td>slash:rr(24)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>25</td>
<td>transitive*</td>
</tr>
<tr>
<td>26</td>
<td>transitive*</td>
</tr>
<tr>
<td>heat(27)</td>
<td>transitive</td>
</tr>
<tr>
<td>watch(28)</td>
<td>transitive</td>
</tr>
<tr>
<td>shove(29)</td>
<td>transitive</td>
</tr>
<tr>
<td>shove:rr(30)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>eat(31)</td>
<td>transitive</td>
</tr>
<tr>
<td>32</td>
<td>transitive</td>
</tr>
<tr>
<td>33 (rr of 32)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>say/do(34)</td>
<td>used with indirect objects only</td>
</tr>
<tr>
<td>35</td>
<td>transitive</td>
</tr>
<tr>
<td>watch:rr(36)</td>
<td>reflexive/reciprocal</td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Murrinh-Patha classifier stems and their assumed valency. Glosses for the classifier stems have been adapted from Blythe et al. (2007) and Nordlinger (2008, 2011, 2015). Assumed valencies marked with * are only tentative as these classifier stems are listed with less than 10 lexical stems in the database.
For the classifier stems 37 and 38, not even a tendency can be established whether these classifier stems should be considered transitive or reflexive/reciprocal. The reason for this lies in the fact that the classifier stems 37 and 38 are not listed in Street (1989) at all. Blythe et al. (2007) provide paradigms for these classifier stems and some examples. However, these examples always contain the reflexive/reciprocal marker -nu, which can combine with reflexive/reciprocal classifier stems as well as with transitive classifier stems. (3.10) provides an example involving the classifier stem 38. Reflexive/reciprocal classifier stems will be discussed in greater detail in Section 3.6.

(3.10) thungku ningam-nu-ninha-nu-birr
    NCfire  3SGS.38.NFUT-RR-DU.M.RR-shoot
    'The two men shot each other.' (Blythe et al. 2007)

For the classifier stems 37 and 38, as well as for the classifier stems which do not occur often in the database, more work is needed to determine their valency (and their semantic content) without doubt.

Finally, the classifier stem feet(7) is listed as ambiguously transitive and intransitive in Table 3.1, as it occurs very often in transitive as well as in intransitive complex predicates. Section 3.5 discusses this classifier stem in more detail.

When trying to determine the valency of lexical stems, the same problem arises that arose for the classifier stems that cannot form a predicate on their own. Lexical stems can never occur on their own without a classifier stem in Murrinh-Patha. Their valency thus also has to be established by the combinatory possibilities with classifier stems and the resulting complex predicate. If lexical stems are most often used in transitive complex predicates, they are considered to be transitive while they are considered to be intransitive if they form intransitive complex predicates most often.

This diagnostics, however, does not work as well for lexical stems as it works for classifier stems. Most classifier stems combine with a large number of different lexical stems, so that it is relatively straightforward to establish in which combination they most often occur in. In contrast, lexical stems usually combine with only a small set of classifier stems or even with only one classifier stem. Sometimes, the meaning of the lexical stem does not become apparent because the different complex predicates that are formed with the same lexical stem but with different classifier stems are very different semantically. In these cases, it is difficult to determine the valency of the lexical stem. For this reason, lexical stems are only tentatively grouped into different valency classes.

As the valency of classifier stems and lexical stems is defined based on the valency of the complex predicates in which they most often occur in, it automatically follows that in most cases, the valency of the classifier stem...
and the lexical stem match in complex predicate formation. Examples of transitive complex predicates, formed with transitive classifier stems and transitive lexical stems, have been presented in (3.7) already. (3.11) provides examples of intransitive complex predicates which involve intransitive classifier stems and intransitive lexical stems.

(3.11)  

a. ngani-ngkamumuy-nu  
\text{1SGS.BE(4).FUT-be.blind-FUT}  
'I'll be blind.' \hspace{1cm} (Street 1989)

b. kanam-kaykay  
\text{3SGS.BE(4).NFUT-call.out}  
'He continually calls out.' \hspace{1cm} (Street 1989)

To sum up, this section discussed how valency can be established for Murrinh-Patha complex predicates as a whole as well as for individual classifier and lexical stems. It was shown that while it is relatively straightforward to assign a valency to the complex predicate as a whole, identifying the valency of the individual classifier and lexical stems is problematic as most classifier stems and all lexical stems cannot function as the sole verbal predicate. This means that the valency of the classifier stems and the lexical stems can only be assigned based on the valency of the complex predicates they occur in. This definition then leads to the fact that in most cases, the valency of the classifier stem and the lexical stem that form a complex predicate match. However, this is not always the case as will be seen in the following sections which discuss the additional patterns that can be found in Murrinh-Patha. While valency is helpful in discussing these various alternation patterns, the difficulties in establishing the valency of classifier and lexical stems show that an argument structure based approach to Murrinh-Patha complex predicate formation is difficult.

### 3.2 Causatives and anticausatives

The previous section discussed how the valency of classifier and lexical stems may be established and showed that based on the definition assumed in this work, valency matching of classifier and lexical stems is the most common pattern of complex predicate formation in Murrinh-Patha. However, other patterns also exist which will be discussed in the remainder of this chapter. This section is concerned with cases in which the valency of classifier stems and lexical stems do not match but for which the number of arguments of the complex predicate as a whole is determined by the number of arguments of the classifier stem.

Intransitive lexical stems can combine with transitive classifier stems to form causative complex predicates in Murrinh-Patha. This pattern can be found in Ngan'gityemerri as well, as was discussed in Section 2.4.5. Similarly,
Schultze-Berndt (2000) reports that in Jaminjung, intransitive coverbs can combine with transitive inflecting verbs and that in these combinations, the argument of the coverb can be identified with either of the arguments of the inflecting verb.

(3.12) provides a Murrinh-Patha example. The lexical stem *dharday* ‘descend’ can combine with the classifier stems 1 to 7 to encode intransitive complex predicates. The lexical stem can thus be considered intransitive. In (3.12a), it combines with the classifier stem *STAND(3)*. It can also combine with the classifier stem *HANDS(8)*, as can be seen in (3.12b), in which case it forms a transitive complex predicate with a causative meaning.

(3.12) a. *ngirra-dharday-nu*

1SGS.STAND(3).FUT-descend-FUT
‘I’ll descend straight down.’ (Street 1989)

b. *nanthi karlay kanhi-ka*

NCres fishing net DEM-FOC
*nguma-dharday-degida-nu-neme*
1DU.S.HANDS(8).FUT-descend-again-FUT-PAUC.M
‘We will drop the fishing net down again.’ (Street 1989)

Another example involving the lexical stem *mum* ‘glow’ can be seen in (3.13). In (3.13a), the classifier stem *STAND(3)* is used to encode an intransitive complex predicate. In contrast, in (3.13b), the transitive classifier stem *HANDS(8)* forms a transitive complex predicate with it.

(3.13) a. *thungku pirrim-mum*

NCfire 3SGS.STAND(3).NFUT-glow(RDP)
‘The (many) fires are glowing.’ (Street 1989)

b. *thungku light mam-mum*

NCfire light 1SGS.HANDS(8).NFUT-glow(RDP)
‘I turned the lights on.’ (Street 1989)

In contrast to *dharday* ‘descend’ in (3.12), however, it is not completely clear whether *mum* ‘glow’ should be considered as an intransitive lexical stem as the gloss suggests. *mum* can only combine with the classifier stems *STAND(3)* and *HANDS(8)*, according to the database.

Alternatively, it could also be an instance of a transitive lexical stem used in an anticausative formation. Such a process is attested in Ngan’gityemerri (c.f. Reid 2000) and is present in Murrinh-Patha as well. Some combinations of transitive lexical stems with the intransitive classifier stem *SIT(1)* result in an intransitive complex predicate with an anticausative/resultative reading. Lexical stems such as *lerrkperrk* ‘crush’ and *warnta* ‘split open’ normally combine with transitive classifier stems which denote the kind of action that
leads to the state of being smashed or split open. An example of *lerrkperrk* ‘crush’ with the transitive classifier stem HANDS(8) is given in (3.14a).

(3.14) a. \(kamtum\) mam-lerrkperrk
\(\text{NC}_{\text{anim}}\) egg \(1\text{SGS.HANDS(8).NFUT-crush}\)
‘I crushed the egg with my hand.’ (Seiss & Nordlinger 2010)

b. dim-lerrkperrk
\(3\text{SGS.SIT(1).NFUT-crush}\)
‘It’s smashed.’ (Seiss & Nordlinger 2010)

c. dim-warnta
\(3\text{SGS.SIT(1).NFUT-split.open}\)
‘It’s cracked.’ (Seiss & Nordlinger 2010)

If these lexical stems combine with the intransitive classifier stem SIT(1) such as in (3.14b,c), the agent argument cannot be realized and the complex predicate receives an anticausative reading. This kind of behavior is also attested with the causative lexical stems *buwurr* ‘fracture’, *parl* ‘break’ and *dharrkat* ‘stick’.

In (3.14b,c) it is the agent that cannot be realized which results in an anticausative/resultative reading. It should be briefly noted here that it is also possible that the theme argument is not expressed. For example in (3.15), *dhegdhek* ‘play’ may combine with the transitive classifier stem HANDS(8) in (3.15a) which yields a transitive verbal complex. It can, however, also combine with the intransitive classifiers stem BE(4), forming an intransitive complex predicate with an activity reading, as can be seen in (3.15b).

(3.15) a. pumamka-dhegdhek-ngime
\(3\text{DU.S.HANDS(8).NFUT-play-PAUC.F}\)
‘They’re playing around with that girl/boy.’ (Seiss & Nordlinger 2010)

b. parnamka-dhegdhek-ngime
\(3\text{DU.S.BE(4).NFUT-play-PAUC.F}\)
‘They’re playing.’ (Seiss & Nordlinger 2010)

These cases of causative and anticausative complex predicates share with the argument structure mapping cases that the number of arguments of the complex predicate is determined by the number of arguments of the classifier stem. However, for a formal analysis they may pose different challenges.

As Schultze-Berndt (2000) shows, cases in which intransitive lexical stems combine with transitive classifier stems to form transitive complex predicates can be easily accommodated in an approach in which the arguments of classifier and lexical stems are identified in complex predicate

---

formation. If only one argument is provided by the lexical stem, one of the two arguments of the classifier stem is identified with the argument of the lexical stem. This can be modeled in Wilson’s (1999) analysis of Wagiman, too.

The anticausative complex predicates might pose more difficult problems for a formal analysis. In these cases, one argument of the lexical stem cannot be expressed openly because the classifier stem only has one argument with which the argument of the lexical stem could be identified. The existing approaches to Australian coverb constructions do not offer such an argument structure operation. See Section 4.1 for a more detailed discussion of this pattern and the difficulties it poses for a formal analysis.

These difficulties with the formal analysis in addition to the problem of not being able to determine the argument structure of lexical stems properly give a hint that having the argument structure of classifier and lexical stems at the heart of a complex predicate analysis for Murrinh-Patha is difficult. Further evidence for this claim is presented in the following sections in which the number of arguments of the complex predicate does not follow the number of arguments of the classifier stems as laid out in Table 3.1.

### 3.3 Intransitive classifier stems in transitive complex predicates

The previous two sections presented combinations of Murrinh-Patha classifier and lexical stems in which the number of arguments of the complex predicate was determined by the valency of the classifier stem. In addition to these patterns, the number of arguments of complex predicates as a whole may not follow the valency of the classifier stems in certain combinations. This section is concerned with intransitive classifier stems in transitive complex predicates while the following Section 3.4 discusses transitive classifier stems in intransitive complex predicates.

These cases are problematic as they cannot be explained with the usual patterns that the approaches such as Wilson (1999) or Schultze-Berndt (2000) propose. Although these patterns can be found in e.g., Wagiman or Jaminjung, Wilson (1999) and Schultze-Berndt (2000) treat these cases as exceptions that require a different set of rules. The approach presented here argues that these cases are actually no isolated exceptions in Murrinh-Patha, but that they are predictable based on the semantic concepts. An approach that covers these cases along with the valency matching cases is therefore to be preferred to an approach in which these cases are treated as pure exceptions. Chapter 4 will present such an approach.

Before looking at Murrinh-Patha, a brief look is cast on the relevant Ngan’gitemerri data. Reid (2000) points out that intransitive classifier stems can combine with what he calls bivalent low transitive lexical stems
in Ngan’gityemerri. He claims that in this case, the focus of the sentence is on the subject movement or activity. An example has been provided in (2.62) already. (3.16) provides another example.

(3.16) \textit{wi-ribem(∅)-tiptip ngan’gi nyinyi} 3SGS/A-\textit{stand(-3sg.do)}-\textit{pick.up} word \textit{2sg} ‘It (tape recorder) is recording your speech.’

(Ngan’gityemerri, Reid 2000:345)

Reid (2000) notes that it is quite difficult to decide whether these complex predicates should be considered transitive or intransitive. Just as in Murrinh-Patha, the third person singular direct object is not overtly expressed in Ngan’gityemerri. According to Reid (2000), these constructions never occur with first or second person direct object marking, either. This means that they never occur with overtly expressed direct object marking. On the other hand, these constructions allow an overtly expressed noun phrase which seems to function as an object, e.g., \textit{ngan’gi nyinyi} ‘your speech’ in (3.16). This makes it difficult to decide whether these constructions should be considered transitive or intransitive. Reid (2000) argues that they should most probably be considered intransitive and overtly expressed noun phrases should be treated as “cognate objects” (Reid 2000:246). He considers these constructions to be ‘less transitive’ in a Hopper & Thompson (1980) sense.

Similar constructions can also be found in Murrinh-Patha. In (3.17), an intransitive classifier stem combines with a transitive lexical stem and co-occurs with an overt noun phrase.

(3.17) a. \textit{kura patha kanamkurdugurduk} kura patha kanam-gurdugurduk
\textit{NC_{water} good 3SGS.BE(4).NFUT-drink(RDP)} ‘He continually drinks water.’

b. \textit{ku ngurlmirl wurrak-ku}
\textit{NC_{anim} fish 3SGS.GO(6).NFUT-fish} ‘He continually catches fish.’

(Street 1989)

Seiss & Nordlinger (2010) treat these examples similarly to the Ngan’gityemerri examples. They consider the noun phrases to be non-referential. In their view, “the noun phrase functions to characterize the activity instead of defining a participant (cf. Van Valin & LaPolla 1997:148ff)” (Seiss & Nordlinger 2010:426), e.g., the activity involved is water-drinking or fish-catching. They thus propose to treat these constructions as (pseudo) noun incorporation as discussed by e.g., Mohanan (1995), Ball (2004) and Duncan (2007). Nordlinger (2011) makes a similar claim for these Murrinh-Patha
constructions in that she argues for a continuum of transitivity and treats these constructions as deviations from the prototypical transitive verbs.

To test this view, a more detailed look is cast on the data. In such constructions, the noun can be modified by adjectives such as terert ‘many’ and ngala ‘big’, as can be seen in (3.18). This shows that if this was an instance of pseudo incorporation, it is most likely one of noun phrase incorporation, as discussed by Dayal (2011) for Hindi and Massam (2001) for Niuean.

(3.18) a. mi terert kanamurrk
    mi terert kanam-murrk
    N_{veg} many 3SGS.BE(4).NFUT-eat
    ‘She’s eating a lot (of apples).’  (Fieldnotes R. Nordlinger)

b. mi ngala kanamurrk
    mi ngala kanam-murrk
    N_{veg} big 3SGS.BE(4).NFUT-eat
    ‘She’s eating a big (apple).’  (Fieldnotes R. Nordlinger)

While the examples above can all be interpreted as involving pseudo noun (phrase) incorporation, some (albeit infrequent) examples exist which suggest that the noun phrase is indeed a direct object. In (3.19a), the noun phrase is clearly referential. In (3.19b,c), the intransitive classifier stems stand(3) and be(4) are used in transitive complex predicate with overtly expressed direct objects. In these cases, then, it is clearer that it is not an instance of noun incorporation.

(3.19) a. mi kanamurrk apple
    mi kanam-murrk apple
    N_{veg} 3SGS.BE(4).NFUT-eat apple
    ‘She’s eating the apple (the one that I left on the table).’
    (Fieldnotes R. Nordlinger)

b. kumparra warra punni-dha berematha
    first first 3PL.S.FEET(7).PIMPF-PIMPF that’s all
    gathu warda
    towards after that
    pirrim-pun-mardaputh
    3SGS.STAND(3).NFUT-3PL.DO-load.up.truck
    ‘A big mob went in front, after that he picked them all up on the truck.’
    (Seiss & Nordlinger 2010)

c. ngani-nan-part-nu-warra
    1SGS.BE(4).FUT-2PL.DO-leave-FUT-now
    ngurru-warra
    1SGS.GO(6).FUT-now
    ‘I’ve got to leave you behind, I’m going.’
    (Seiss & Nordlinger 2010)
The empirical facts remain to be fully determined in terms of which combinations of intransitive classifier stems and lexical stems can combine with direct objects in these ways, and under what conditions. However, the intransitive classifier stems that are involved in such constructions are classifier stems of posture and motion like sit(1), lie(2), stand(3) or go(6).

What these classifier stems contribute is often not very clear. For example, in (3.20), the lexical stem mirrmirr ‘thunder rumbling’ combines with the classifier stems sit(1), be(4) or go(6). In (3.20a) and (3.20b), the meaning of the complex predicate is the same, whether the lexical stem is combined with the classifier stem sit(1) or be(4). This means that the meaning of sit(1) as denoting a specific posture is probably bleached in this case. In contrast, the classifier stem go(6) which is used in (3.20c) seems to contribute some of its original meaning as denoting a sound coming from the distance in this case.

(3.20) a. dim-mirrmirr
   3SGS.sit(1).nfut-rumble
   ‘There’s a big noise (like thunder).’ (Fieldnotes R. Nordlinger)

b. kanam-mirrmirr
   3SGS.be(4).nfut-rumble
   ‘There’s a big noise (like thunder).’ (Fieldnotes R. Nordlinger)

c. wurran-mirrmirr
   3SGS.go(6).nfut-rumble
   ‘Sound coming from distance, can hear the noise from a
distance.’ (Fieldnotes R. Nordlinger)

Street (1996) argues that the classifier stems 1 to 6 convey aspectual information in addition to posture or stance. Reid (2002) makes a similar claim for Ngan’gityemerri. While this seems to be true for both languages, the details for Murrinh-Patha remain to be established. One open question is whether all classifier stems 1 to 6 can really be used aspectually and what they would contribute in this case. The classifier stems seem to differ in their productivity. That is, some classifier stems, namely the classifier stems sit(1), be(4) and go(6), seem to be able to be used with a very wide range of lexical stems.

As was seen above, the lexical stem mirrmirr ‘thunder rumbling’ can combine with the classifier stems sit(1), be(4) and go(6). (3.21a) shows that it cannot be used with the classifier stem stand(3). Similarly, the lexical stem merrmerr ‘roar(RDP)’, can be used with the classifier stems sit(1), be(4) and go(6), but not with the classifier stem lie(2), which is shown in (3.21b). This provides evidence for the assumption that the classifier stems sit(1), be(4) and go(6) are more productive and are probably able to convey aspectual information exclusively, i.e., without contributing information about posture or motion.
Murrinh-Patha complex predicate patterns

(3.21)  
\begin{align*}
\text{a. } & \text{pirrim-mirrmir} \\
& 3\text{SGS.STAND}(3).\text{NFUT-\textit{rumble}} \\
& \text{(Fieldnotes R. Nordlinger)} \\
\text{b. } & \text{yibim-merrmerr} \\
& 3\text{SGS.LIE}(2).\text{NFUT-\textit{roar(RDP)}} \\
& \text{(Fieldnotes R. Nordlinger)}
\end{align*}

A second open question is the relation between the aspectual information contributed by the classifier stems and aspectual specifications that are required by the lexical stem. As was discussed in Section 1.1, lexical stems can be reduplicated in Murrinh-Patha. Reduplicated lexical stems seem to contribute aspectual information and have an effect on the combinatory possibilities with classifier stems. For example, as can be seen in (3.22), the lexical stem \textit{merr} ‘roar’ cannot combine with the classifier stem \textit{go}(6), but the reduplicated form \textit{merrmerr} can. The exact semantics of the reduplication and the interplay with the classifier stem remain to be established.

(3.22)  
\begin{align*}
\text{a. } & \text{wurran-merrmerr} \\
& 3\text{SGS.GO}(6).\text{NFUT-\textit{roar(RDP)}} \\
& \text{‘It (thunder, waves) is roaring.’} \\
& \text{(Fieldnotes R. Nordlinger)} \\
\text{b. } & \text{*wurran-merr} \\
& 3\text{SGS.GO}(6).\text{NFUT-\textit{roar}} \\
& \text{(Fieldnotes R. Nordlinger)}
\end{align*}

The intransitive posture and motion classifier stems can also be used in another construction in Murrinh-Patha (and in Ngan’gityemerrir, cf. Reid 2002). The classifier stems can be used as serialized clitics cliticizing onto a classifier and lexical stem combination. In this case, they yield different tense and aspect specifications (cf. Street 1996, Nordlinger & Caudal 2012), as can be seen in (3.23). As Nordlinger & Caudal (2012) point out, (3.23b), which involves a complex predicate in non-future tense with the serialized classifier stem \textit{sit}(1), is present imperfective in contrast to (3.23a), which has a past perfective reading.

(3.23)  
\begin{align*}
\text{a. } & \text{dirraninthanubath} \\
& \text{dirran-nintha-nu-bath} \\
& 3\text{SGS.WATCH}(28).\text{NFUT-DU.M-RR-look.at} \\
& \text{‘They (two) looked at each other.’} \\
\text{b. } & \text{dirraninthanubath=dim} \\
& \text{dirran-nintha-nu-bath=dim} \\
& 3\text{SGS.WATCH}(28).\text{NFUT-DU.M-RR-look.at=}3\text{SGS.SIT}(1).\text{NFUT} \\
& \text{‘They (two) are looking at each other.’} \\
& \text{(Nordlinger & Caudal 2012:85)}
\end{align*}

It thus seems that the posture and motion classifier stems can be quite bleached of their meaning, at least when they are used as serialized clitics.
Their use as the sole classifier stem in complex predicates still needs to be more detailedly studied, but most probably they can be very bleached in these cases as well.

The grammaticalization path from posture and motion verbs to tense and aspect markers is well documented across languages, e.g., Heine (1993), Kuteva (2001), Lemmens (2005) or Juge (2006). It should therefore not be surprising to find bleaching in Murrinh-Patha posture and motion classifier stems as well. This then facilitates their use in transitive complex predicates.

These classifier stems can thus combine quite freely with intransitive as well as with transitive lexical stems and the valency of the complex predicate follows the valency of the lexical stem. In this case, the argument structure does not play a role. Instead, the semantic content of the classifier stems as contributing mainly aspeccual information enables the combination with transitive as well as intransitive lexical stems. It seems to depend on the lexical stem whether it can be used with this kind of intransitive classifier stem. The lexical stems that can be used with these intransitive classifier stems do not need a classifier stem that adds more than aspeccual meaning to the complex predicate. In a way, these lexical stems can function as the verbal predicate on their own semantically, but they need a classifier stem to fulfill their morpho-syntactic requirements such as person and tense marking.

Schultze-Berndt (2000) even tentatively interprets similar cases in Jamijung as auxiliary plus verb constructions in which the inflecting verb functions as the auxiliary and the coverb functions as verb. However, Baker & Harvey (2010) point out that the evidence for considering these combinations as auxiliary verb constructions while treating other coverb constructions as complex predicates is not convincing. They conclude that it is very difficult to draw a distinction between auxiliary verb interpretation and a complex predicate interpretation for Australian coverb constructions.

Wilson (1999) considers similar constructions in Wagiman as complex predicates, but proposes a different formalism for the analysis. While he proposes that other coverb constructions can be modeled by merging the LCS of the coverb into the LCS of the inflecting verb, he assumes that the inflecting verb ‘-ya ‘go’ has another LCS which embeds the full LCS of the coverb in its LCS as an XFUNC, as was discussed in Section 2.4.1.

Other approaches to Australian coverb constructions thus treat this use of intransitive classifier stems in transitive complex predicates as exceptions, i.e., as instances of pseudo noun incorporation, auxiliary verb constructions or by adding a new formalism. The data found in Murrinh-Patha suggests that it is not an instance of pseudo-noun incorporation. The examples thus really involve transitive complex predicates. As the approach proposed in Chapter 4 does not put the argument structure at the heart of the analysis, these cases are not considered as exceptional as in the other approaches.
3.4 Transitive classifier stems in intransitive complex predicates

The previous section dealt with intransitive classifier stems that could be used in transitive complex predicates. This section now considers cases in which transitive classifier stems are used in intransitive complex predicates, providing examples from the classifier stems HANDS(8) and POKE(23). In these combinations, the transitive classifier stems combine with intransitive lexical stems to form intransitive complex predicates. As has been pointed out in Chapter 2, this pattern is rather uncommon in Australian languages, but it is attested in Jaminjung (Schultze-Berndt 2000) and Bardi (Bowern 2004). In contrast to Jaminjung and Bardi, however, no overt object marker is present in the Murrinh-Patha examples.

The pattern discussed in this section contrasts with the cases of causative complex predicates in which transitive classifier stems combine with intransitive lexical stems to form transitive complex predicates, discussed in Section 3.2. It thus depends on the combination of classifier and lexical stem whether the resulting complex predicate is transitive or intransitive. In the causative complex predicates, the lexical stem denotes a result state for the event that is specified by the classifier stem. In contrast, if transitive classifier stems are used in intransitive complex predicates, one meaning part of the classifier stem licenses the use of the classifier stem in these combinations.

The classifier stem HANDS(8) is discussed in detail by Barone-Nugent (2008). He describes the prototypical sense of the classifier stem as “x is in physical contact over a period of time with y using hands” (Barone-Nugent 2008:53). He provides a range of examples in which the classifier stem HANDS(8) contrasts with other classifier stems in this sense. One example is provided in (3.24), in which the classifier stem HANDS(8) contrasts with the classifier stem BASH(14) in transitive complex predicates.

(3.24) a. \( \text{ku} \quad \text{tumtum} \quad \text{mam-lerrkperrk} \nC_{\text{anim}} \text{egg} \quad 1\text{SG.HANDS(8).NFUT-crack} \)

‘I cracked the egg with my hands.’ (Barone-Nugent 2008)

b. \( \text{ku} \quad \text{tumtum} \quad \text{bangam-lerrkperrk} \nC_{\text{anim}} \text{egg} \quad 1\text{SG.BASH(14).NFUT-crack} \)

‘I cracked the egg with a rock.’ (Barone-Nugent 2008)

Barone-Nugent (2008) shows that the classifier stem HANDS(8) has a range of other meanings that can be derived from this prototypical meaning. Especially important for the discussion at hand is the use of the classifier stem in intransitive complex predicates such as in (3.25) which involve gliding events.
Murrinh-Patha complex predicate patterns

(3.25)  
a.  *ma-wel-nu*  
3SG.HANDS(8).FUT-glide-FUT  
‘It will glide.’  
(Barone-Nugent 2008:55)

b.  *mam-peyway*  
3SG.HANDS(8).NFUT-glide  
‘It glided.’  
(Barone-Nugent 2008:56)

c.  *ma-dhayway-nu*  
2SG.HANDS(8).FUT-glide.in.water-FUT  
‘It will fly amongst the waves.’  
(Barone-Nugent 2008:56)

Barone-Nugent (2008) argues that the classifier stem HANDS(8) can be used with these lexical stems because fins and wings are considered as hand-like body parts and hands are an important part of the prototypical meaning of the classifier stem. This then facilitates the use of the classifier stem HANDS(8) in these complex predicates denoting gliding events even though the event is intransitive and not transitive.3

A similar behavior can also be found with the classifier stem POKE(19). It has also been studied in detail by Barone-Nugent (2008), who states that the basic meaning of POKE(19) is that of events in which contact is made with a pointed end of an instrument, such as a stick or spear. This use of the classifier stem is illustrated in (3.26).

(3.26)  
ku  
thithay  
nganthak=ngem  
ku  
thithay  
ngam-thak=ngem  
NC_anim  
honey  
1SG.S.POKE(19).NFUT-dip=1SG.S.SIT(1).NFUT  
‘I’m dipping into the honey.’  
(Street 1989)

Beyond this basic meaning, POKE(19) can also be used in intransitive events. Barone-Nugent (2008) shows that POKE(19) can be used in events involving linear movement such as in (3.27a,b). He also points out that POKE(19) can be used in events that denote plant growth such as in (3.27c).

(3.27)  
a.  *nga-wintigat-nu*  
1SG.S.POKE(19).FUT-descend-FUT  
‘I’m going down.’  
(Fieldnotes R. Nordlinger)

b.  *lalingkin pa-rturt-nu*  
sea  
3SG.S.POKE(19).FUT-advance-FUT  
‘The sea will advance (tide come in).’  
(Barone-Nugent 2008)

3 It is crucially assumed that these complex predicates are indeed intransitive complex predicates, and that they do not denote a reflexive meaning of moving oneself along with the help of hand-like body parts. If this reading was indeed correct, I would expect that the reflexive/reciprocal classifier stem HANDS:RR(10) (discussed in Section 3.6) is used in these complex predicates. I thank Boban Arsenijevic for bringing up this point.
c. nanthi parnu
   NC_{res} grass
dam-pithpitj=wurran
3SG.POKE(19).NFUT-sprout=3SG.GO(6).NFUT
‘The grass is sprouting.’ (Barone-Nugent 2008)

He explains the use of the classifier stem in the examples involving movement and growth by assuming that linear movement is an integral part of the prototypical use of the classifier stem POKE(19), so that the classifier stem can be used in events in which the movement is actually not brought about by an external source. That is, the prototypical use of POKE(19) involves an agent that controls the linear movement of a long pointed object, but the linear movement is so important for the prototypical use that it alone can license the use of the classifier stem in non-prototypical, intransitive complex predicates as well.

The classifier stems HANDS(8) and POKE(19) are thus two examples of transitive classifier stems used in intransitive complex predicates. They can be considered transitive classifier stems because they mostly form transitive complex predicates. However, if the semantic meaning allows it, they can also be used in intransitive complex predicates. Most importantly, these cases are not idiosyncratic, rare exceptions, but actually show a certain degree of systematicity. The classifier stem HANDS(8) selects for a range of lexical stems with the meaning ‘glide’. Similarly, the classifier stem POKE(19) is used in different complex predicates of linear movement. Further transitive classifier stems, e.g., BASH(14) and SLASH(23), also form intransitive complex predicates in predictable contexts (see Section 4.2 for details).

Because the classifier stems each pose their own requirements on the semantics of the lexical stem they can combine with, their behavior in these complex predicates is predictable and not idiosyncratic. In an account in which the argument structure is the most prominent criterion to predict the combinatory possibilities, these examples would have to be treated as purely exceptional along the lines of really idiosyncratic combinations.

3.5 The classifier stem FEET(7): An example of an ambiguous classifier stem

The previous subsections discussed clearly intransitive classifier stems used in transitive complex predicates and transitive classifier stems used in intransitive complex predicates. For these classifier stems it could be argued that they are either intransitive or transitive based on the meaning that they contribute and based on the transitivity of the complex predicates they most often form. This section discusses the classifier stem FEET(7) for which it is very difficult to establish whether it should be considered transitive or
intransitive. The section actually argues that this classifier stem is best analyzed as ambiguously transitive and intransitive.

The classifier stem FEET(7) can be used in combinations with lexical stems or, along with the intransitive classifier stems 1 to 6, it can function as the sole verbal predicate. When used alone, it is intransitive and denotes movement, as can be seen in (3.28).

(3.28) a. ngunungam-warda
    1SGS.FEET(7).NFUT-now
    ‘I’m going now.’ (Nordlinger 2011)

b. kunungam pangu-warda-da
    3SGS.FEET(7).EXIST distal-now-place
    ‘He’s gone off now.’ (Fieldnotes R. Nordlinger)

However, in combination with lexical stems it may either be transitive or intransitive. This can be seen in the examples in (3.29). In (3.29a), the lexical stem thurr ‘tread’ combines with the classifier stem FEET(7) to encode a transitive complex predicate. In contrast, in (3.29b), the lexical stem rdardarr ‘hobble’ combines with FEET(7) to encode an intransitive complex predicate.

(3.29) a. ngunu-thurr-nu
    1SGS.FEET(7).FUT-tread.on-FUT
    ‘I’ll tread on it.’ (Street 1989)

b. punu-rdardarr-nu
    3SGS.FEET(7).FUT-hobble-FUT
    ‘He will hobble.’ (Street 1989)

In its transitive use, the classifier stem FEET(7) denotes an action carried out with the foot. In this case, it often contrasts with the classifier stem HANDS(8) and many minimal pairs such as in (3.30) can be found.

(3.30) a. ngunu-dhardum-nu
    1SGS.FEET(7).FUT-press-FUT
    ‘I’ll press it down with my foot.’ (Street 1989)

b. manthardum
    mam-dhardum
    1SGS.HANDS(8).NFUT-press
    ‘I pressed it down.’ (Street 1989)

In the intransitive use of the classifier stem FEET(7), movement is the main factor that plays a role in the selection process. It may involve movement by feet such as in (3.31a), but can also be used when the feet are not involved
in the movement process such as in (3.32a). In this use, the classifier stem FEET(7) mainly patterns with the other intransitive classifier stems denoting posture and movement, i.e., classifier stems 1 to 6. That is, many lexical stems that can combine with the classifier stems 1 to 6 are also possible with the classifier stem FEET(7). (3.31) and (3.32) provide examples for the lexical stems bangurr ‘white’ and barlbarl ‘fly’ which can combine with the classifier stem FEET(7) and with the classifier stems 1 to 6. In (3.31), the lexical stem bangurr ‘white’ combines with the classifier stems FEET(7) and with the classifier stems 1 to 6. In (3.31), the lexical stem bangurr ‘white’ combines with the classifier stems FEET(7) and sit(1), which illustrates that FEET(7) contributes movement to the complex predicate while sit(1) contributes a sitting posture. (3.32) involves the lexical stem barlbarl ‘fly’ and the classifier stems FEET(7) and GO(6) which shows that it is actually very difficult to determine the difference in meaning between the intransitive use of FEET(7) and GO(6).

(3.31) a. nguunu-bangurr-nu
1SGS.FEET(7).FUT-white-FUT
‘I’ll go dressed in white.’ (Street 1989)
b. dimmanangurr
dim-bangurr
3SGS.sit(1).NFUT-white
‘He’s sitting in white.’ (Street 1989)

(3.32) a. nungammarlbarl
nungam-barlbarl
3SGS.FEET(7).NFUT-fly
‘It is flying.’ (Street 1989)
a. purru-barlbarl-nu
3SGS.GO(6).FUT-fly-FUT
‘It will fly.’ (Street 1989)

To summarize the discussion about the classifier stem FEET(7), if it is used as the sole verbal predicate, it is intransitive. In some combinations of classifier and lexical stems, the classifier stem FEET(7) is intransitive and denotes movement. However, it also combines with lexical stems in transitive combinations in which it denotes actions carried out by foot. I therefore assume that the classifier stem FEET(7) should be considered ambiguous transitive and intransitive.

Ambiguous classifier stems can of course be accommodated in approaches that use the argument structure as a first means to explain the combinatorial possibilities of classifier and lexical stems. However, in such approaches they are a special case. In the approach advocated for in this thesis, i.e., in which semantic concepts denoted by the classifier stem determine the combinatorial possibilities and the argument structure alternations are derived from these
combinations, such ambiguities are actually expected and they do not form an exceptional case. The only special point about the classifier stem FEET(7) in this view is that there is not really a prototypical usage of the classifier stem from which the other usages can be derived. Instead, the classifier stem FEET(7) is very common both in transitive and intransitive complex predicates.

3.6 Reflexive/reciprocal classifier stems

The previous sections discussed transitive and intransitive classifier stems, but Murrinh-Patha also has reflexive/reciprocal (RR) classifier stems. Complex predicates formed with these RR classifier stems are the topic of this section. Similarly to the intransitive and transitive classifier stems in the previous sections, RR classifier stems can be used in complex predicates with different argument structure frames. As would be expected, they are used in RR complex predicates. However, they can also form intransitive complex predicates. That RR classifier stems can be used in non-reflexive and non-reciprocal complex predicates has been already discussed by Nordlinger (2008). This section adds to this view by considering the interaction of reflexivization/reciprocalization and the applicative marker -ma which illustrates the difference between the RR and intransitive complex predicates formed with RR classifier stems clearly.

As the glosses provided for the RR classifier stems indicate, some transitive classifier stems have corresponding RR classifier stems, i.e., transitive lexical stems which combine with a transitive classifier stem may also combine with the corresponding RR classifier stem. For example, (3.33a) shows a transitive combination of the classifier stem HANDS(8) and the lexical stem kurrk ‘scratch’. In (3.33b), the corresponding RR classifier stem HANDS:RR(10) is used with the same lexical stem which triggers a reflexive meaning.

(3.33) a. mam-kurrk
    1SG.HANDS(8).NFUT-scratch
    ‘I scratched it.’ (Street 1989)

b. mem-kurrk
    1SG.HANDS:RR(10).NFUT-scratch
    ‘I scratched myself.’ (Street 1989)

However, as was pointed out by Nordlinger (2008) and can be seen in Table 3.2, some RR classifier stems actually function as RR correspondences to two different non-RR classifier stems. As Nordlinger (2012a) argues, this is indicative of a purely formal correspondence between the RR and

---

4This section is mainly taken from Seiss & Nordlinger (2010).
non-rr classifier stems instead of one that is driven by meaning. An example of such a complex correspondence is provided by the classifier stem bash:rr(15), which functions as rr correspondence for the classifier stems see(13) and bash(14). Another such correspondence involves the classifier stem hands:rr(10) which functions as rr correspondence for hands(8) and snatch(9). An example of the alternation of hands:rr(10) and snatch(9) is provided in (3.34).

\begin{align*}
(3.34) & \quad a. \textit{mangan-tha} \\
& \quad 1\text{sgS}.\text{snatch}(9).\text{nfut}-\text{chase} \\
& \quad \text{‘I chased him.’} & \text{(Street 1989)} \\
& \quad b. \textit{pume-tha-nu} \\
& \quad 1\text{inclS}.\text{hands:rr}(10).\text{fut}-\text{chase-fut} \\
& \quad \text{‘We (incl) will fight.’} & \text{(Street 1989)}
\end{align*}

As can be seen from (3.34), rr classifier stems may also trigger a reciprocal meaning when the subject is in non-singular form. Another example is given in (3.35b), involving the classifier stem slash(23) and its rr correspondence slash:rr(24).

\begin{align*}
(3.35) & \quad a. \textit{ngu-nhi-bat-nu} \\
& \quad 1\text{sgS}.\text{slash}(23).\text{fut}-2\text{sg}.\text{do-hit-fut} \\
& \quad \text{‘I’m going to hit you.’} & \text{(Seiss & Nordlinger 2010)} \\
& \quad b. \textit{puy-bat-nu} \\
& \quad 1\text{inclS}.\text{slash:rr}(24).\text{fut-hit-fut} \\
& \quad \text{‘We are going to hit each other.’} & \text{(Seiss & Nordlinger 2010)}
\end{align*}

\footnote{The semantic difference between reflexives and reciprocals is not considered here as reflexives and reciprocals will be treated alike from an argument structure point of view.}
Examples as provided in (3.33) to (3.35) are straightforward, productive combinations of RR classifier and lexical stems. Their combinations are usually predictable based on the combinatory possibilities of their corresponding non-RR classifier stems. Additionally, the resulting complex predicate is reflexive or reciprocal.

RR classifier stems plus lexical stem combinations may also result in non-reflexive or non-reciprocal meanings. Lexical stems like *wengkawuy* ‘confuse’ or *mardat* ‘amaze’ can be used with transitive classifier stems in experiencer-object complex predicates (along the lines of Croft (1991:214)) as in (3.36).

(3.36)  
\[ \text{a. pan-ngi-wengkawuy} \]  
\[ 3\text{SGS.SLASH(23).NFUT-1SG.DO-confuse} \]  
‘He confused me.’  
(Street 1989)  
\[ \text{b. ma-nhi-mardat-nu} \]  
\[ 1\text{SGS.HANDS(8).FUT-2SG.DO-amaze-FUT} \]  
‘I’ll amaze you.’  
(Street 1989)

When these lexical stems are used with the corresponding RR classifier stems as in (3.37), the resulting complex predicate is not reflexive or reciprocal, i.e., the combination does not result in a coindexation of the arguments involved. Rather, the combination denotes an intransitive result state, e.g., in (3.37a), the speaker reports a state of being confused, in which the speaker is not necessarily the source of the confusion himself. Similarly, in (3.37b), the source of the amazement are fish, not the men themselves, which would be the case if we treat the RR classifier stem as coindexing the arguments of the complex predicate.

(3.37)  
\[ \text{a. ngurdampengkawuy} \]  
\[ \text{ngurdam-wengkawuy} \]  
\[ 1\text{SGS.SHOVE:RR(30).NFUT-confuse} \]  
‘I’m confused.’  
(Street 1989)  
\[ \text{b kardu ngamere-ka pumem-mardat} \]  
\[ \text{NC\textsubscript{human} few-FOC 3PLS.HANDS:RR(10).NFUT-amaze} \]  
\[ \text{ku ngurlmirl nhini-nu-yu} \]  
\[ \text{NC\textsubscript{anim} fish DEM-DAT-DM} \]  
‘And the few men were amazed at all those fish.’  
(Street 1989)

While *wengkawuy* ‘confuse’ and *mardat* ‘amaze’ can be used with both RR and non-RR classifier stems, a range of lexical stems exists which can only occur with RR classifier stems. This is the case with the lexical stems *nham* ‘fear’ and *ngkabat* ‘surprise’, illustrated in (3.38). These lexical stems cannot combine with other classifier stems (as far as the database is concerned).

(3.38)  
\[ \text{a. nhem-nham} \]  
\[ 1\text{SGS.POKE:RR(21).NFUT-fear} \]  
‘I’m afraid.’  
(Seiss & Nordlinger 2010)
In contrast to the productive RR classifier plus lexical stem combinations discussed above, the combinations in (3.37) and (3.38) do not involve RR complex predicates with coindexed arguments. They rather denote intransitive complex predicates involving an experiencer subject. They can be considered semi-productive and predictable as it seems that the lexical stem determines whether the arguments are coindexed or whether the resulting complex predicate is intransitive. For example, (3.37) and (3.38) suggests that emotion lexical stems with an experiencer object result in an intransitive complex predicate if they combine with an RR classifier stem. Further evidence for this view is provided in the following by considering the crosslinguistic perspective and the interaction of the RR classifier stems with another RR marker -nu and the applicative marker -ma.

That RR markers are used in non-RR constructions like in (3.37) and (3.38) is quite common crosslinguistically. The Murrinh-Patha examples resemble what Steinbach (2002) calls German inherent reflexive constructions ((3.39)) in which the reflexive pronoun is not a semantic argument of the verb, but is needed syntactically.

(3.39) Er fürchtet sich.
   He be.afraid self.acc
   ‘He is afraid.’

Similarly, Kemmer (1993) treats reflexive markers in constructions like the German and Murrinh-Patha examples as middle markers which should be given a semantic analysis distinct from reflexives. She considers several sub-classes of verbs in which reflexive markers are often used as middle markers crosslinguistically. Besides body care verbs and different verbs of motion, emotion verbs like the Murrinh-Patha examples are named as prototypical verb classes in which reflexive markers are used as middle markers. Thus, the Murrinh-Patha examples in (3.37) and (3.38) fit into the crosslinguistic tendencies which allow reflexive markers to be used in intransitive, stative events.

Evidence that different argument structures can be encoded with RR classifier stems also comes from the interaction of the RR classifier stems with the RR marker -nu (or nunggu for paucal subjects). As is laid out in Nordlinger (2008), the RR marker -nu always coindexes the arguments involved. This leads to argument structure alternations that show that the RR classifier stems either form RR complex predicates or intransitive complex predicates. To show this, the basic use of the RR marker -nu as well as its interaction with the RR classifier stems is discussed in the following.
Murrinh-Patha complex predicate patterns

As Nordlinger (2008) notes, -nu is used to encode a reflexive/reciprocal meaning for classifier plus lexical stem combinations for which no corresponding RR classifier stems exist. An example of such an alternation involving the classifier stem FEET(7) is provided in (3.40b) for the non-reflexive/reciprocal combination in (3.40a).

(3.40)  

(a) nungarnirtnda  
nungam-rirda  
3SGS.FEET(7).NFUT-push  
‘He kicked him.’  
(Nordlinger 2008)

(b) nungam-ngintha-nu-rirda=pirrim  
3SGS.FEET(7).NFUT-DU.F-RR-push=3SGS.STAND(3).NFUT  
‘They kicked each other.’  
(Nordlinger 2008)

Additionally, -nu can combine with RR classifier stems such as in (3.41). As was pointed out by Nordlinger (2008), the presence of -nu in combination with RR classifier stems seems to trigger no or only little change in meaning in most cases.

(3.41)  

(a) pam-nintha-thuk  
pam-nintha-nu-thuk  
3SGS.SLASH:RR(24).NFUT-DU.M-fight  
‘They two fought each other.’  
(Nordlinger 2008)

(b) pam-nintha-nu-thuk  
3SGS.SLASH:RR(24).NFUT-DU.M-RR-fight  
‘They two fought each other.’  
(Nordlinger 2008)

However, in some cases a difference in meaning is apparent, as is shown in Nordlinger (2008) and illustrated with the examples in (3.42). These are the cases in which the RR classifier stem does not coindex the arguments provided by the lexical stem but instead results in an intransitive result state. As can be seen in (3.42a), this not only happens for experiencer-object complex predicates but also for complex predicates which involve lexical stems like let ‘stick’. This means that different subgroups of lexical stems exist which show this behavior when combined with RR classifier stems.

(3.42)  

(a) mem-let  
3SGS.HANDS:RR(10).NFUT-stick  
‘It’s already stuck (e.g. on the wall).’  
(Nordlinger 2008)

(b) mem-nu-let  
3SGS.HANDS:RR(10).NFUT-RR-stick  
‘It stuck itself (e.g. on the wall).’  
(Nordlinger 2008)
The complex predicate in (3.42a) which just consists of the RR classifier stem and the lexical stem let ‘stick’ has a stative, resultative meaning. In contrast, if -nu is added as in (3.42b), the resulting complex predicate receives a reflexive reading. As is pointed out in Nordlinger (2008), this shows that the RR morphological marker -nu only has the RR function, i.e., that -nu always coindexes the two arguments of the lexical stem, while the RR classifier stems are ambiguous between both uses.

Similar alternations can also be found when considering the interaction of RR classifier stems with the applicative marker -ma. This marker attaches to the verbal complex in the slot for incorporated body parts. The applicative promotes a source to the function of a direct object as can be seen in (3.43) to (3.45). (3.43a) shows a verbal complex with an oblique argument specifying the source while (3.43b) shows the applicativized version in which the source ‘from me’ has been promoted to the function of a direct object which is now marked on the verbal complex by the first person singular direct object marker ngi. (3.44) is a similar example with the second person direct object marker nhi marking the source, while (3.45) shows that the original object can still be expressed as an overt noun phrase kurə ‘water’.

(3.43)  
\begin{align*}
a. & \quad \text{T}ruck \ darrarart \ pumangan-art \ ngarra \ ngay. \\
& \quad \text{truck stolen 3PLS.SNATCH(9).NFUT-get LOC 1SG} \\
& \quad \text{‘They stole a truck from me.’} \quad (\text{Seiss & Nordlinger 2010}) \\
b. & \quad \text{pumangan-ngi-ma-art} \\
& \quad \text{3PLS.SNATCH(9).NFUT-1SG.DO-APPL-get} \\
& \quad \text{‘They took it from me.’} \quad (\text{Seiss & Nordlinger 2010})
\end{align*}

(3.44) nganam-nhi-ma-kut  
\begin{align*}
& \quad \text{1SGS.BE(4).NFUT-2SG.DO-APPL-collect} \\
& \quad \text{‘I collected (the money) from you.’} \quad (\text{Seiss & Nordlinger 2010})
\end{align*}

(3.45) mangan-nhi-ma-art  
\begin{align*}
& \quad \text{kura} \\
& \quad \text{1SGS.SNATCH(9).NFUT-2SG.DO-APPL-get NC\text{\textsubscript{water}}} \\
& \quad \text{‘I got (some) water from you.’} \quad (\text{Seiss & Nordlinger 2010})
\end{align*}

This applicative marker can now function as a further test to show that the argument structure of complex predicates formed with RR classifier stems can either involve two coindexed arguments or a single argument. This can be illustrated with the contrast in (3.46) and (3.47).

In the RR combination in (3.46) involving the RR classifier stem slash:RR(24) and the lexical stem rartal ‘cut off’, the source is coindexed

---

6See Nordlinger (2009) for justification of the applicative analysis, and discussion of the relationship between the applicative marker and the incorporated body part -ma ‘hand’.
with the agent in both sentences, independent of whether -nu is present; in other words, in both cases the reciprocal relation holds between the tearers and those the clothes were torn from.

(3.46)  

a. pam-ngintha-ma-rartal  
3SGS.SLASH:RR(24).NFUT-DU.F-APPL-CUT.off(RDP)  
‘They tore the clothes from each other.’  
(Seiss & Nordlinger 2010)

b. pam-ngintha-nu-ma-rartal  
3SGS.SLASH:RR(24).NFUT-DU.F-RR-APPL-CUT.off(RDP)  
‘They tore the clothes from each other.’  
(Seiss & Nordlinger 2010)

(3.47) involves the RR classifier stem POKE:RR(21) and the lexical stem nham ‘fear’. As can be seen in (3.47a), the RR classifier stem results in an intransitive complex predicate as in (3.37) above. The source of the fear is not the subject itself in this case. In (3.47b), the second person direct object marker nhi is added as a source by the applicative marker -ma. In this case, there is no reflexive/reciprocal relation holding between the subject and applied object argument. In order to express such a relation, the RR marker -nu needs to be added, as in (3.47c). This then results in a coindexation of the arguments involved.

(3.47)  

a. ngem-nham  
1SGS.POKE:RR(21).NFUT-fear  
‘I’m afraid.’  
(Seiss & Nordlinger 2010)

b. ngem-nhi-ma-nham  
1SGS.POKE:RR(21).NFUT-2SG.DO-APPL-fear  
‘I’m afraid of you.’  
(Seiss & Nordlinger 2010)

c. them-nu-ma-nham  
1INCLS.POKE:RR(21).NFUT-RR-APPL-fear.  
‘We’re (inclusive) frightened of each other.’  
(Seiss & Nordlinger 2010)

The contrast in (3.46) and (3.47) shows that RR classifier stems can either be used in complex predicates in which the arguments are coindexed or in intransitive complex predicates. In complex predicates in which the arguments are coindexed as in (3.46), no difference in meaning occurs whether -nu is present. This is also the case if an applicative is involved. In contrast, in (3.47), a difference in meaning is apparent if -nu is present. Without -nu, the RR classifier stem results in an intransitive complex predicate describing the state of being in fear. If a source of this fear is added and if that source should be coindexed with the subject, then the RR marker -nu has to be added.
Thus, the interaction with the applicative marker -ma and the RR marker -nu again shows that RR classifier stems can be used in RR complex predicates in which the arguments are coindexed or in intransitive complex predicates that denote a stative, often resultative event. The RR complex predicates are the productive, predictable combinations, but also the intransitive complex predicates formed with RR classifier stems are predictable to a certain degree, e.g., for combinations of experimenter-object lexical stems in combination with RR classifier stems.

3.7 Conclusion

In this chapter, the various Murrinh-Patha complex predicate patterns were discussed from an argument structure point of view. The chapter laid out how the valency of classifier and lexical stems can be determined. It was argued that the valency of classifier and lexical stems has to be based on the valency of the complex predicates they most often occur in. Although this might be difficult, especially for lexical stems, it was argued that no alternative exists. However, the difficulties with establishing the valency of lexical stems show that an argument structure approach to complex predicate formation is problematic.

Under the assumed definition of the valency of classifier and lexical stems, the most common pattern to complex predicate formation is that of argument matching: transitive classifier stems combine with transitive lexical stems to form transitive complex predicates while intransitive classifier stems combine with intransitive lexical stems to form intransitive complex predicates.

However, in addition to this most frequent pattern, a range of other patterns can be found which make an analysis that concentrates on the argument structure in the selection process of classifier and lexical stems difficult. It was demonstrated that in Murrinh-Patha, intransitive classifier stems can occur in transitive complex predicates and transitive classifier stems can occur in intransitive complex predicates. It could be shown that these are not purely idiosyncratic, exceptional combinations but that these combinations are to a certain degree predictable. For example, the classifier stem HANDS(8) is productively used in transitive complex predicates in which an action is carried out with the hands. However, the same classifier stem can be used in intransitive complex predicates in which hands or hand-like body parts play a role as well (cf. Barone-Nugent 2008).

Similarly, it could be shown that the RR classifier stems, in addition to their most common use in RR complex predicates, very productively form intransitive complex predicates if the semantics of the lexical stems allow this. Finally, the classifier stem FEET(7) was shown to be ambiguously transitive and intransitive.
What this discussion shows is that an analysis which puts the argument structure as a first criterion to determine whether classifier and lexical stems are compatible cannot account straightforwardly for all of the data. The analysis proposed in the following chapter thus does not make use of the argument structure of the classifier and lexical stems as used in the discussions in this chapter.
Murrinh-Patha complex predicate patterns
Chapter 4

An LCS blueprint analysis of complex predicates

This chapter presents a semantic analysis for the Murrinh-Patha classifier and lexical stem combinations. It argues for an account using LCS blueprints for classes of complex predicates which specify which information is contributed by the classifier stem and which information is contributed by the lexical stem. For example, in (4.1), the complex predicate is a causative in which the result state is contributed by the lexical stem *mel* ‘flatten’ and the instrument is contributed by the classifier stem HANDS(8).

\[(4.1) \quad \text{mam-mel} \]
\[
\text{1SGS.HANDS(8).NFUT-flatten} \]
\[\text{‘I flattened it out by hand’ (Street 1989)} \]

While LCSs have been used in a wide range of works on complex predicates, for example by Butt (1995), Andrews & Manning (1999), Wilson (1999), Broadwell (2000), Nordlinger (2010b) or Baker & Harvey (2010), they have not been used as blueprints which model the behavior of whole classes of complex predicates. In these earlier approaches, LCSs are used to restrict the combinatory possibilities in complex predicate formation, i.e., two elements can only form a complex predicate if their LCSs are compatible. Although the rules for compatibility can be defined for a language individually, I argue in this chapter that these approaches cannot easily be applied to the Murrinh-Patha data. As was discussed in Chapter 3, argument structure does not play a prominent role in the selection process of Murrinh-Patha complex predicates. However, according to these previous proposals, argument structure would be central for an LCS analysis of Murrinh-Patha complex predicates.

While argument structure does not restrict the combinatory possibilities of Murrinh-Patha classifier and lexical stem combinations, the combinations
are not unrestricted either. Chapter 3 already briefly discussed that semantic concepts such as a specific type of movement or the involvement of a specific body part restrict the combinatory possibilities. This chapter is concerned with these semantic concepts in more detail. A case study of the classifier stems bash(14), poke(19) and slash(23) shows that each classifier stem may cover a range of different semantic concepts which pose selectional restrictions on the combinations with lexical stems. The various combinatory possibilities can be displayed in different hierarchies of selectional restrictions.

These hierarchies of selectional restrictions can be used to derive the argument structure of the complex predicates. For the modeling of the argument structure, LCSs are used as they mirror the restrictions of the productive and semi-productive combinations modeled in the hierarchy of selectional restrictions. For example, they offer the possibility of incorporating the shape of an object denoted by the classifier stem, which restricts the combinatory possibilities of caused contact complex predicates. LCSs can thus be used to model the argument structure of complex predicates once the combinatory possibilities are established by the hierarchies of selectional restrictions.

This analysis thus combines two quite distinct approaches to lexical semantics, namely a primitive-based and a relation-based approach to lexical semantics. Following Pustejovsky (1995), primitive-based approaches (e.g., Wilks 1975, Katz 1972, Jackendoff 1990) try to decompose the meaning of a word into a certain set of primitives while relation-based approaches (e.g., Quillian 1968, Fodor 1975, Asher 2011) model meaning by a network of links between different words. This chapter argues that both levels are needed in the modeling of Murrinh-Patha complex predicates.

The chapter is structured as follows. Section 4.1 provides an overview over previous approaches to complex predicates that are relevant for the LCS blueprint analysis of Murrinh-Patha complex predicates. It briefly summarizes previous LCS based approaches and discusses why it is difficult to adapt such approaches to Murrinh-Patha complex predicates. Additionally, it presents and evaluates Caudal et al.’s (2013) approach to Murrinh-Patha complex predicates.

Section 4.2 presents a case study of bash(14), poke(19) and slash(23) and illustrates the hierarchies of selectional restrictions for these classifier stems. The LCS blueprint analysis is presented in Section 4.3. The section argues for modeling the argument structure of Murrinh-Patha complex predicates with LCSs, introduces the basic idea of the blueprint analysis and illustrates how LCS blueprints are linked to the hierarchies of selectional restrictions. Section 4.4 concludes the chapter.
4.1 Previous approaches

This section provides a brief overview over previous approaches to complex predicates that are relevant for the LCS blueprint analysis of Murrinh-Patha classifier and lexical stem combinations presented in this chapter. The section first summarizes previous LCS based approaches and discusses difficulties in their adaptation to the Murrinh-Patha data (Section 4.1.1). Section 4.1.2 then presents the analysis proposed by Caudal et al. (2013a) for Murrinh-Patha complex predicates.

4.1.1 Previous LCS based approaches and difficulties in the adaptation to Murrinh-Patha

Many approaches to complex predicate formation use Jackendoff’s (1990) LCSs to model the compatibility of the complex predicate constituents and to exclude illformed combinations. These analyses were inspired by Butt’s (1995) analysis of complex predicates in Urdu. For Australian languages, LCSs have been used in the analysis of complex predicates in e.g., Wagiman (Wilson 1999, Wilson 2006, Andrews & Manning 1999), Wambaya (Nordlinger 2010b) or across languages (Baker & Harvey 2010). Butt’s (1995) account of Urdu complex predicates as well as Wilson’s (1999) and Baker and Harvey’s (2010) account of Australian complex predicates have been discussed in detail in Chapter 2. This section briefly summarizes these approaches and discusses why such approaches are difficult to adapt to Murrinh-Patha complex predicates. It is argued that Murrinh-Patha complex predicates display too much variety in their combinatory possibility to explain complex predicate formation with a restricted set of LCS rules.

Butt (1995) is concerned with complex predicates in Urdu and other languages in which so-called light verbs, which roughly correspond to Murrinh-Patha classifier stems, combine with a main verb, noun or adjective. To model the semantic contribution of each part of the complex predicate, Butt (1995) uses LCSs for each of the parts and different mechanisms of how these LCSs can combine. She accounts for the fact that light verbs are semantically bleached by proposing an LCS for the light verb which contains a transparent event. In complex predicate formation, the LCS of the full verb is inserted into the transparent event and, depending on the light verb, either event fusion or argument fusion takes place.

Wilson (1999) uses the idea of LCSs for complex predicate formation in the Australian language Wagiman. However, he shows that a different approach is needed in Wagiman, as the two parts which form a complex predicate in Wagiman can combine in more diverse ways than in Urdu. In contrast to Butt’s (1995) approach to Urdu complex predicates involving a transparent event, the LCS of the coverb is merged into the LCS of the inflecting verb wherever it fits. This accounts for the fact that coverbs
An LCS blueprint analysis of complex predicates

and inflecting verbs can combine in various ways in Wagiman, and for the observation that inflecting verbs in Wagiman do not necessarily have to be semantically light. Ungrammatical combinations are ruled out if merging the LCS of the coverb into the LCS of the inflecting verb is impossible.

Additionally, Wilson (1999) proposes a second rule of combining LCSs for aspectual uses of the inflecting verb -ya- ‘GO’. An example has been provided in (2.48) and is repeated in (4.2).

(4.2) 
gartgart-da ga-ya yimbama

’he is always laughing’ (Wagiman, Wilson 1999:157)

Wilson (1999) proposes to model this use of the inflecting verb -ya- ‘GO’ with an LCS that contains a placeholder entry XFUNC that specifies where the LCS of the coverb should be merged into. As can be seen in (4.3), the LCS contains two main predicates, GO and XFUNC and their arguments are coindexed (specified by α).

(4.3) 
-ya- ‘GO’

\[
\text{Event} \text{GO} ([\text{Thing}]) \quad \text{WITH} \quad ([\text{XFUNC}(\alpha)])
\]

Wilson (1999) thus proposes two different rules for complex predicate formation in Wagiman. For most combinations, he assumes a merger account in which the LCS of the coverb is merged into the LCS of the inflecting verb. The rule for the aspectual use of -ya- ‘GO’, in contrast, is a coindexation account: the LCSs are not merged, but the arguments are coindexed.

The distinction between coindexation and merger complex predicates has been discussed in more detail by Baker & Harvey (2010). They claim that most Australian coverb constructions are merger constructions while Ngangityemerri and its neighbors, i.e., also Murrinh-Patha, display coindexation patterns. They base this claim on combinations of intransitive classifier stems used in associated motion constructions such as in (4.4) for which they propose a coindexation analysis.

(4.4) 
nga-nanim-fifi

1SGS-GO.PR-smoke

’I’m going along smoking.’ (Ngangitymerri, Baker and Harvey 2010)

Baker & Harvey (2010) assume that languages make use of either coindexation or merger constructions. In this view, then, all Murrinh-Patha classifier and lexical stem combinations would be coindexation constructions. This view is not only challenged by the Wagiman data and Wilson’s (1999) analysis, it is also difficult to uphold for the Murrinh-Patha data.

In contrast to merging, coindexation does not restrict the combinatory possibilities straightforwardly. A coindexation approach is thus especially
well suited for combinations without strict restrictions. Wilson (1999) for Wagiman as well as Butt (1995) for Urdu use the coindexation analysis for light verbs which can combine with other elements rather freely to form complex predicates. Butt (1995), for example, proposes to analyze the Urdu permissive with an argument fusion approach, which would be considered a coindexation approach in Baker and Harvey’s (2010) approach. In this construction, the light verb *de* ‘let’ can form a permissive complex predicate with both transitive and intransitive main verbs (Butt 1995:35). In Murrinh-Patha, in contrast, classifier stems can only combine with a restricted set of lexical stems. To avoid overgeneration, a pure coindexation approach for Murrinh-Patha complex predicate formation would have to introduce a large set of additional rules to restrict the combinatory possibilities.

Moreover, Murrinh-Patha complex predicates share many patterns with Wagiman complex predicates, so that applying a merger account along the lines of Wilson (1999) seems to be feasible for these combination. An example is provided in (4.5) and (4.6). In Wagiman, the inflecting verb -bu-‘hit’ can combine with coverbs of impact and violence as in (4.5). The Murrinh-Patha classifier stems slash(23) or bash(14), for example, fulfill very similar functions, as can be seen in (4.6).

(4.5)  
\[
\begin{array}{c}
dalh-ma \\
punch-ASP \\
3SG>1SG-HIT-PST \\
nga-di-n-ma \\
1SG-COME-PRS-DS \\
\end{array} \\
\text{‘he punched me here on the nose and my blood came out.’}
\]
(Wagiman, Wilson 1999, 119)

(4.6)  
\[
\begin{array}{c}
\text{banganthekum} \\
\text{bangam-yekum} \\
\text{3SGS.BASH(14).NFUT-hit.on.ear} \\
\text{‘He hit him on the ear (with stone/spear).’} \\
\text{Street 1989} \\
\text{pan-ngi-yekum} \\
\text{3SGS.SLASH(23).NFUT-hit.on.ear} \\
\text{‘He hit me on the ear (with a stick/small spear).} \text{ (Street 1989)}
\end{array}
\]

If, however, both coindexation and merger constructions are involved in complex predicate formation in Murrinh-Patha, it needs to be established which combinations involve merger constructions and which involve coindexation constructions. In Wagiman and Urdu, this seems to be bound to the light verb that is used. For Murrinh-Patha, this is not as straightforward. The classifier stems 1-6 can be used in associated motion and posture complex predicates as in the Ngan’gityemerri example in (4.4) above. In this case, they seem to involve coindexation constructions. The classifier stem sit(1),
An LCS blueprint analysis of complex predicates

however, is also used in anticausative/resultative constructions as discussed in Chapter 3. An example is repeated in (4.7).

\( (4.7) \) \textit{dim-lerrkperrk} \\
n\text{3SGS.sit(1).NFUT-crush} \\
\text{‘It is smashed.’} \quad \text{(Seiss \& Nordlinger 2010)}

This combination involving the classifier stem \textit{sit(1)} seems to involve a merger approach instead of a coindexation approach as the resulting complex predicate describes a state of being smashed. Whether a coindexation or merger constructions is used in Murrinh-Patha thus depends on the combination of classifier and lexical stem involved, not solely on the classifier stem.

(4.7) provides a further challenge. The classifier stem \textit{sit(1)} was assumed to be an intransitive classifier stem in Chapter 3 and its LCS as a simple stative classifier stem is proposed in (4.8a). Determining the LCS of the lexical stem is more difficult. As was discussed in Chapter 3, the lexical stem \textit{lerrkperrk} ‘crush’ most often combines with transitive, causative classifier stems to form transitive, causative complex predicates. Following the assumptions made in Chapter 3, this means that the LCS for the lexical stem \textit{lerrkperrk} ‘crush’ can be assumed to be as in (4.8b).

(4.8c) displays the LCS of the combined classifier and lexical stem. The resulting complex predicate is an intransitive classifier stem in which the result state is determined by the lexical stem. It seems that the single argument of the classifier stem \textit{sit(1)} picks out the theme object of the lexical stem and thus reduces the number of arguments the lexical stem takes. This cannot be accounted for by the coindexation analysis as coindexation is not able to reduce arguments. However, merging the LCS of the lexical stem in terms of Wilson’s (1999) proposal does not work either.

\( (4.8) \)

a. \textit{sit(1)} \\
\quad \left[ \begin{array}{l} \text{State BE} \\
\quad \left( \begin{array}{l} \text{Thing A}, \text{Place -} \end{array} \right) \end{array} \right] \\

b. \textit{lerrkperrk} ‘crush’ \\
\quad \left[ \begin{array}{l} \text{Event CAUSE} \\
\quad \left( \begin{array}{l} \text{Thing A}, \text{Event BECOME} \\
\quad \left( \begin{array}{l} \text{State BE} \\
\quad \left( \begin{array}{l} \text{Thing A}, \text{Place AT} \\
\quad \left( \begin{array}{l} \text{Property CRUSHED} \end{array} \right) \right) \right) \right) \right) \right] \\

c. \textit{sit(1)-lerrkperrk} ‘be crushed’ \\
\quad \left[ \begin{array}{l} \text{State BE} \\
\quad \left( \begin{array}{l} \text{Thing A}, \text{Place AT} \\
\quad \left( \begin{array}{l} \text{Property CRUSHED} \end{array} \right) \right) \right) \right]

The problem is that in Wilson’s (1999) approach, the LCS of the lexical stem should always be merged into the LCS of the classifier stem. This is not possible in this case as the LCS of the causative lexical stem cannot be merged into the LCS of the stative classifier stem. What happens intuitively is that the LCS of the lexical stem is reduced, i.e., the events \text{CAUSE} and
BECOME in the LCS of the lexical stem cannot be expressed because they do not match with the LCS of the classifier stem. While the process of picking out a patient argument of a lexical stem can be explained in terms of LCSS, the process itself changes the algorithm put forward by Wilson (1999) considerably, as the construction violates monotonicity.

Given that the assumptions about the argument structure and therefore the LCS decomposition is quite unsure for lexical stems as they never function as the sole verbal predicate, I conclude that a different approach is called for. For example, one could assume that the lexical stem *lerkperrk* ‘crush’ is ambiguous between an intransitive and a transitive use. However, this would only solve the problem for this combination. As was seen in Chapter 3, mismatches between the number of arguments of the classifier stem or the lexical stem and the number of arguments of the complex predicate as a whole are quite common. One example, provided already in Chapter 3, is the classifier stem HANDS(8) which can be used in transitive complex predicates in which an action is carried out by hand or in intransitive complex predicates in combination with lexical stems of *gliding* or *flying*. Barone-Nugent (2008) argues that the perception of wings and similar body parts as hand-like entities facilitates the use of the classifier stem in this kind of complex predicate. The two uses are illustrated in (4.9).

\[(4.9)\]

<table>
<thead>
<tr>
<th>Type</th>
<th>LCS</th>
<th>Meaning</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>mam-mel 1SGS.HANDS(8).NFUT-flatten</td>
<td>‘I flattened it out by hand.’ (Street 1989)</td>
<td></td>
</tr>
<tr>
<td>b.</td>
<td>ma-wel-nu 3SGS.HANDS(8).FUT-gliding-FUT</td>
<td>‘It will glide.’ (Barone-Nugent 2008)</td>
<td></td>
</tr>
</tbody>
</table>

The common core meaning that licenses the use of the classifier stem HANDS(8) in both types of complex predicates seems to be that hands or similar body parts play a role. However, this cannot be captured in one LCS that covers both uses as one is a causative complex predicate and the other a complex predicate denoting movement. Two different LCSS would thus be needed for the classifier stem HANDS(8). While this is of course possible, it is problematic for an approach that models the combinatory possibilities purely by the compatibility of the LCSS. Proposing different LCSS for one classifier stem in such an approach bears the risk of overgenerating. At the same time, the approach loses explanatory power if many LCSS are defined for one classifier stem.

To sum up, applying an account along the lines of Wilson (1999) or Baker & Harvey (2010) to the Murrinh-Patha data is difficult. The Murrinh-Patha data does not allow a straightforward division of classifier stems that are used in merger constructions versus coindexation constructions. Although some patterns of combinations of inflecting verbs and coverbs are similar
in Wagiman to the combinations of classifier and lexical stems in Murrinh-Patha, the differences that do exist result in a more complicated system. The combinatorial possibilities of classifier and lexical stems are much higher in Murrinh-Patha than what has been described by Wilson (1999) for Wagiman. This is due to the wide range of meanings which are associated with one classifier stem.

As a consequence, either a different set of combination rules or different LCSs for the same classifier stem would be needed to account for the different combinations. A larger set of rules or different LCSs, however, would not serve to restrict the combinatorial possibilities in Murrinh-Patha complex predicates, they would overgenerate. Different LCSs are needed in many cases as the LCS decomposition is too detailed to account for the combinatorial possibilities. That is, LCSs require the specification of the valency, the path requirements etc., while probably all that the classifier stem HANDS(8), for example, denotes is that it has something to do with hands.

Consequently, an account building on the compatibility of LCSs does not have the necessary explanatory power for Murrinh-Patha. This is not to say that LCSs cannot be helpful in establishing the meaning of a certain range of classifier and lexical stems and their combinations. LCSs offer a way of decomposing the meaning of the complex predicate and show which meaning parts evolve from which element of the complex predicate. In Sections 4.2 and 4.3, an approach is proposed that combines the decomposition of complex predicates into LCSs with a fine-grained semantic hierarchy which is used to model the combinatorial possibilities.

4.1.2 The TCL approach by Caudal et al. (2013a)

Caudal et al. (2013a) propose to model Murrinh-Patha complex predicates within the framework of Type Composition Logic (TCL) (Asher 2011), making use of the notion of coercion to model the combinatorial possibilities. This section briefly highlights some important aspects of the TCL formalism and the adaptation to the Murrinh-Patha data by Caudal et al. (2013a). TCL has also been used to account for verb-formation patterns in the Australian language Panyjima by Caudal et al. (2012).

The main underlying idea of Asher’s (2011) TCL account is that words usually have simple lexical entries which “make simple contributions to truth conditional content” (Asher 2011:22). These simple lexical entries, however, are associated with a range of information of types which guide predication. Asher (2011) uses this system to treat phenomena such as co-predication and coercion in which type clashes occur.

Examples of coercion, taken from Asher (2011:14), are provided in (4.10). In these cases, one word coerces another word to have a different meaning in this particular context. For example, children and lunch in (4.10a) coerce a different meaning for the word good in the different contexts: children can
be referred to as good if they behave well whereas a good lunch is used if
the lunch tastes good. Similarly, the meaning of a cigarette is coerced into
an event reading smoking a cigarette by the verb started in (4.10b). The
coercion is triggered by a type mismatch as start expects an event as its
argument while a cigarette denotes an object.

(4.10) a. good lunch, good children
    b. John started a cigarette.

These and similar cases of coercion have been the matter of extensive re-
search. For example, Pustejovsky (1995) proposed that in such cases of coer-
cion, one specific meaning aspect is selected. The different meaning aspects
are part of a Generative Lexicon and are listed as qualia roles. Pustejovsky
(1995) argues for four different qualia roles for each object. For example,
the telic role specifies the object’s purpose or function. This qualia role is
coerced in (4.10b) as the purpose of a cigarette is that it is being smoked.

Asher (2011) extends the Generative Lexicon approach to handle data
more flexibly. He does not restrict his approach to a small set of roles that
can be coerced. Additionally, in his approach it is not only the object of
the verb which triggers different event readings but the choices can also be
restricted by the subject. In (4.11), if the subject who enjoyed the book is
human, it is most likely that the event was one of reading a book. Alterna-
tively, if the subject is a goat, then it is most probably an eating event.

(4.11) a. Mary enjoyed the book.
    b. The goat enjoyed the book.

Asher (2011) proposes that such specifications are part of the lexicon and are
modeled as bridging functions which are needed if a type clash occurs. For
example, (4.12a) specifies that if there is a subject $\alpha$ which is a human and
there is an object $\beta$ which has a “physical” and an “informational” aspect $(P \bullet I)$ such as a book, then it follows for a statement involving coercion $(\text{EV}(\alpha, \epsilon(\alpha, \beta)))$ that the event $(\epsilon)$ in which the subject and the object are
involved is most probably a reading event. (4.12b) accounts for the fact that
the event is most probably an eating event if the subject is a goat.

(4.12) Bridging functions (Asher 2011:228):

a. $(\alpha \subseteq \text{HUMAN} \land \beta \subseteq P \bullet I) \rightarrow (\text{EV}(\alpha, \epsilon(\alpha, \beta))) > \epsilon(\alpha, \beta) = \text{READ}(\alpha, \beta)$

b. $(\alpha \subseteq \text{GOAT} \land \beta \subseteq P \bullet I) \rightarrow (\text{EV}(\alpha, \epsilon(\alpha, \beta))) > \epsilon(\alpha, \beta) = \text{EAT}(\alpha, \beta)$
The $>$ is used as a weak conditional operator accounting for the defeasibility of the rule. This is important as the interpretation depends on the context. For example, if (4.11b) is uttered in the context of a fairy tale in which goats can read, the rule in (4.12b) might be overridden and the goat might actually also enjoy reading the book. For modeling the context sensitivity, Asher (2011) proposes to combine TCL with Segmented Discourse Representation Theory (Asher & Lascarides 2003).

Caudal et al. (2013a) present a first draft of a TCL analysis of Murrinh-Patha complex predicates. They propose that each complex predicate is of an additive, conjunctive type (⊕) for which the additive conjunction operator $\land_{\oplus}$ combines the underspecified event predicate types of the classifier stem and the lexical stem. If the classifier and lexical stems are of the same type, this operator combines the argument and event structure of the classifier and lexical stem. If, on the other hand, a type clash occurs between the classifier and lexical stem, a bridging function may coerce the required types. If such a bridging function can be defined, the operator $\land_{\oplus}$ proceeds with the combining of the classifier and lexical stem. If no such bridging function can be defined, the combination of classifier and lexical stem is ungrammatical.

Caudal et al. (2013a) use the example in (4.13) to illustrate their approach. (4.14a) displays the starting point of the derivation, i.e., before complex predicate formation takes place. It specifies underspecified event predicate types for the classifier stem ($\phi_1$) and the lexical stem ($\phi_2$). $\pi_1$ and $\pi_2$ are special arguments assumed in TCL which carry the inherent semantic typing restrictions of predicative elements i.e., they model the typing restrictions of the classifier stem and the lexical stem in this case.

The interpretation of the event predicate types depends on their arguments, e.g., $\phi_1$ has two human participants, and is of type bash. The participants carry the restriction that the first argument, i.e., the subject, needs to be of type agent while the second argument, the direct object, needs to be of type patient. In contrast, the lexical stem event predicate type $\phi_2$ is of type hit and requires three participants (HUMAN, HUMAN, PHYS-OBJ). The participants need to be an agent, a patient and an instrument.

(4.13) *banganthekum*
*bangam-yekum*
3SGS.BASH(14).NFUT-hit.on.ear

‘He hit him on the ear (with stone/spear).’ (Street 1989)

(4.14) a. $\lambda w, x, y, z, e_1, e_2, \pi_1, \pi_2 \phi_1 \in \langle BASH, HUMAN, HUMAN \rangle (e_1, v, w, \pi_1)$
$\land_{\oplus} \phi_2 \in \langle HIT, HUMAN, HUMAN, PHYS-OBJ \rangle (e_2, x, y, z, \pi_2)$
$\land \pi_1 * ARG^0_1 : AGENT, ARG^0_1 : PATIENT$
$\land \pi_2 * ARG^0_2 : AGENT, ARG^0_2 : PATIENT, ARG^0_3 : INSTRUMENT$
\[ \lambda x, y, z, e, \pi \phi_1 \in (BASH, HUMAN, HUMAN)(e, x, y, z) \]
\[ \land \phi_2 \in (HIT, HUMAN, HUMAN, PHYS-OBJ)(e, x, y, z) \]
\[ \land \pi \ast ARG_1 : AGENT, ARG_2 : PATIENT, ARG_3 : INSTRUMENT \]

In complex predicate formation, the additive conjunction operator \(\land\) tries to combine the argument and event structure of \(\phi_1\) and \(\phi_2\). As this fails, a bridging function is applied that ‘lifts up’ \(\phi_1\) to receive an instrument argument. \(\land\) then proceeds to combine \(\phi_1\) and \(\phi_2\). The result of the complex predicate formation is provided in (4.14b). The combination now has a joint argument \(\pi\) which specifies the semantic typing restrictions of the complex predicate as a whole, i.e., the complex predicate requires an agent, a patient and an instrument as its arguments. The combination now also shares the same event argument \(e\).

Caudal et al. (2013a) claim that this approach is more flexible than LCS based approaches as it allows for a scale from fully integrated to partially integrated complex predicates, in contrast to the binary distinction of merger or coindexation constructions of the LCS based approaches. That is, they assume that in complex predicate formation, \(\phi_1\) and \(\phi_2\) should at least share the argument \(\pi\) which reflects that they are formally unified. In fully integrated complex predicates, the event structure, argument structure and overall semantics are shared while in partially integrated types the classifier and lexical stem may retain their argument and event structures. This thus is one advantage of the TCL approach over the LCS based approaches.

A further advantage is the context sensitivity that can be modeled with TCL. Although Caudal et al. (2013a) do not make use of it, TCL as proposed by Asher (2011) offers ways of incorporating context sensitivity through defeasible bridging functions and a link to Segmented Discourse Representation Theory (Asher & Lascarides 2003). Context sensitivity may be important in Murrinh-Patha, for example, for complex predicates involving the intransitive motion and posture classifier stems. As was discussed in Chapter 3, these classifier stems can be very bleached of their meaning so that it is not clear whether they contribute motion or posture. An example is provided in (4.15), in which the classifier stems are used aspectually.

\(4.15\)

a. \textit{ngem-bukka}  
\[
1SG.SIT(1).NFUT-have.red.eyes
\]
‘I have bloodshot eyes.’ (Street 1987)

b. \textit{nukunu-ka wurranmukka}  
\[
nukunu-ka wurran-bukka
3SG.M-FOC 3SG.GO(6).NFUT-have.red.eyes
\]
‘He continually has red/bloodshot eyes.’ (Street 1987)
As Nordlinger (p.c.) points out, in an unmarked context the classifier stems are most likely interpreted aspectually. In certain contexts however, for example in contrastive contexts, the same combination of classifier and lexical stems may result in a posture or motion interpretation. The TCL approach could assume an underspecified lexical entry for these classifier stems which triggers the required interpretation in the appropriate context. In contrast, LCS approaches have to assume two distinct LCSs for the two interpretations, one involving merger and one involving coindexation, and do not have a worked out mechanism of how to choose between the two analyses.

However, the TCL approach as proposed by Caudal et al. (2013a) also has its shortcomings. Just as the LCS approaches discussed in the previous Section 4.1, the argument structure (and event structure) of the classifier and lexical stems plays a prominent role. However, as was discussed in Chapter 3, determining the argument structure of classifier stems and especially of lexical stems is not straightforward in Murrinh-Patha.

The idea of type mismatches and coercion at first glance seems to be attractive to distinguish between very productive complex predicate combinations and more constrained combinations. As was discussed in Chapter 3, some combinations of classifier and lexical stems are more productive and should thus be considered more basic and straightforward than others. Barone-Nugent (2008), for example, even speaks of prototypical combinations. For example, the classifier stem \texttt{hands(8)} occurs much more often in transitive complex predicates of caused contact than in intransitive complex predicates of movement. In principle, the coercion functions seem to be attractive to model such differences. The intuition would be that productive combinations do not need coercion while other combinations need coercion. For example, the classifier stem \texttt{hands(8)} in a transitive complex predicate would not need coercion but \texttt{hands(8)} in intransitive complex predicates of movement would need coercion.

Unfortunately, however, the division is not as straightforward in the approach proposed by Caudal et al. (2013a). As the types that they use are very fine-grained, type clashes occur even in productive complex predicates. For example, they assume a coercion function for the straightforward, productive complex predicate in (4.13). Thus, the intuition that some combinations of classifier and lexical stems are more straightforward and easily accommodated than others is not captured with the distinction between complex predicate formation which needs coercion versus complex predicate formation that does not need coercion.

Finally, the TCL approach does not offer a decomposition of the lexical semantics of the complex predicate as a whole in terms of which meaning aspect is provided by the classifier stem and which meaning aspect is provided by the lexical stem. Such a decomposition is incorporated in the LCS approaches and is also included in approaches in First Phase Syntax such as Ramchand (2008b). TCL does not assume that such a composition is necessary.
However, the Murrinh-Patha data suggests that in many cases the classifier stem and the lexical stems indeed contribute different aspects of meaning to the complex predicate. For example, the classifier stem may contribute the specific shape of an instrument while the lexical stem may contribute the result state to the overall semantics of the complex predicate.

The following section discusses the various aspects of meaning that can be contributed by the classifier stem and the lexical stem and its restrictions on the combinations for a case study involving the classifier stems BASH(14), POKE(19) and SLASH(23).

4.2 Hierarchies of selectional restrictions for BASH(14), POKE(19) and SLASH(23)

This section is concerned with a case study of complex predicates formed with the classifier stems BASH(14), POKE(19) and SLASH(23) and the selectional restrictions that these classifier stems pose. The classifier stem POKE(19) has been briefly discussed from an argument structure perspective in Chapter 3, which showed that POKE(19) can be used in transitive as well as in intransitive complex predicates. This is also true for the classifier stems BASH(14) and SLASH(23). Nevertheless, complex predicate formation with these classifier stems is not unrestricted. This section is concerned with the finer-grained semantic concepts that pose restrictions on the combinatorial possibilities of these classifier stems with lexical stems.

For POKE(19), Barone-Nugent (2008) has already presented a detailed study, cast within cognitive semantics. His findings are used here to contrast them with the behavior of the classifier stems BASH(14) and SLASH(23). For the purpose of determining the combinatorial possibilities and selectional restrictions of BASH(14) and SLASH(23), I sorted the lexical stems which combine with these classifier stems into different semantic classes. Street (1989) has been used as the main database for this task, complemented with fieldnotes by Rachel Nordlinger. By sorting through the combinations of this database, it can be shown that all classifier stems, in addition to occurring in transitive complex predicates involving caused contact, can occur in complex predicates involving movement and mouth-associated actions.

While these different meaning dimensions are thus shared by all three classifier stems, finer-grained semantic distinctions exist which restrict the combinatorial possibilities further. This means that not all lexical stems can combine with the three classifier stems BASH(14), POKE(19) and SLASH(23) as the semantic concepts denoted by the classifier stems and the meaning of the lexical stems have to be compatible, or, in other words, the lexical stem and the classifier stem pose selectional restrictions on each other.

1 An earlier version of this section was published as Seiss (2012a).
Thus, while some lexical stems can combine with all three classifier stems bash(14), poke(19) and slash(23) or even more classifier stems, some lexical stems can only combine with two or one of the classifier stems. These combinations can be ordered hierarchically, with lexical stems that combine with a large set of classifier stems on top of the hierarchy and complex predicates in which the lexical stem only combines with one classifier stem on the bottom of the hierarchy. As this hierarchical ordering of the combination also implicitly reflects the selectional restrictions that the classifier and lexical stems pose on each other, the ordering is called “hierarchies of selectional restrictions”.

### 4.2.1 Complex predicates of caused contact

Barone-Nugent (2008) states that the basic meaning of poke(19) is that of events in which contact is made with a pointed end of an instrument, such as a stick or spear. This use of the classifier stem is illustrated in (4.16).

(4.16) ku thithay nganthak=ngem
ku thithay ngam-thak=ngem
NCAnim honey 1SG.POKER(19).NfUT-dip=1SG.SIT(1).NfUT
‘I’m dipping into the honey.’ (Street 1989)

In this use of caused contact, the classifier stem poke(19) contrasts with the classifier stems bash(14) and slash(23). bash(14) is used to denote events in which flat, solid objects such as stones, hammers etc. play a role. In contrast, slash(23) denotes events in which the long side of an object such as a knife etc. figures prominently. (4.17) provides an example of the contrastive use of the lexical stem parrang ‘numb’ with all three classifier stems.

(4.17) a. nga-nhi-ma-parrang-nu
1SG.Poke(19).FUT-2SG.DO-IBP-numb-FUT
‘I’ll numb your hand (by injection).’ (Street 1989)

b. bangam-parrang
1SG.Bash(14).NfUT-numb
‘I made him numb (with stone/spear).’ (Street 1989)

c. ngu-nhi-me-parrang-nu
1SG.Slash(23).FUT-2SG.DO-IBP-numb-FUT
‘I’ll numb your foot (with stick).’ (Street 1989)

The same contrast can be seen in (4.18) involving the lexical stems yekum ‘hit on ear’ and its reduplicated form yemkum. When combining with bash(14), the action is carried out with a stone or spear while when combining with slash(23), the action is carried out with a stick or small
An LCS blueprint analysis of complex predicates

spear. (4.18b) involves the classifier stem POKE:RR(21), which is the reflexive/reciprocal version of POKE(19) and which denotes the action of poking a stick in one’s ear, generally for cleaning purposes (Street 1989).

(4.18) a. bangantehekum
    bangam-yekum
    3SGS.BASH(14).NFUT-hit.on.ear
    ‘He hit him on the ear (with stone/spear).’
    (Street 1989)

b. denthemkum=wurran
    dem-yemkum=wurran
    3SGS.POKE:RR(21).NFUT-hit.on.ear(RDP)=3SGS.GO(6).NFUT
    ‘He is always poking (sticks) in his ear.’
    (Street 1989)

c. pan-ngi-yekum
    3SGS.SLASH(23).NFUT-hit.on.ear
    ‘He hit me on the ear (with a stick).’
    (Street 1989)

However, not all lexical stems can combine with all three different classifier stems to form complex predicates in which caused contact plays a role. The shape of the instrument poses selectional restrictions on the lexical stem, i.e., the classifier stem and the lexical stem have to be compatible so that for example the shape of the instrument can bring about the result state denoted by the lexical stem.

For example, the lexical stem mel ‘flatten’ can only combine with the classifier stems BASH(14) and SLASH(23), but not with the classifier stem POKE(19). An action in which something is flattened with the tip of a long pointed instrument is probably very rare, so that the combination is not attested in the database. In this way, the shape of the instrument denoted by the classifier stem restricts the combinatory possibilities with the lexical stem.

(4.19) a. bangam-melmel=wurran
    3SGS.BASH(14).NFUT-flatten(RDP)=3SGS.GO(6).NFUT
    ‘He continually flattens it out.’
    (Street 1989)

b. ngu-mel-nu
    1SGS.SLASH(23).FUT-flatten-FUT
    ‘I will flatten it out (with a stick/pipe).’
    (Street 1989)

When searching through the database, many of these finer-grained restrictions can be found. Lexical stems of hitting often also just combine with the classifier stems BASH(14) and SLASH(23). One example is provided in (4.20) with the lexical stem rlip ‘hit’. Another lexical stem that patterns in the same manner is dheng ‘hit heavily’.
An LCS blueprint analysis of complex predicates

(4.20) a. panperlip
    pan-we-rlip
    3SG.SLASH(23).NFUT-IBP-hit
    ‘He hit him on the head (with a stick).’ (Street 1989)

b. thu palyirr-te bangam-ngi-we-rlip
    NC_weapon stone-INSTR 3SG.BASH(14).NFUT-1SG.DO-IBP-hit
    ‘He hit me on the head with a stone.’ (Street 1989)

Finally, a range of lexical stems of hitting only combine with one of the classifier stems in question, depending on the fine-grained lexical semantics the lexical stem denotes. For example, the lexical stem *rde* ‘hit with something pointed or round (e.g., short spear, fist, hammer, bullet, blunt end of axe, etc.)’ can only combine with the classifier stem *bash(14)*, according to Street (1989). It thus seems that the lexical semantics of the lexical stem *rde* incorporate the same concepts as the classifier stem *bash(14)* so that it can only combine with this classifier stem.

(4.21) be-nderde-dha=ngardi
    1SG.BASH(14).PIMPF-hit(RDP)-PIMPF=1SG.BE(4).PIMPF
    ‘I was hitting it.’ (Street 1989)

Similarly, the lexical stem *bat* ‘hit once’ is only listed with the classifier stem *slash(23)*, as can be seen in (4.22). Street (1989) specifies that this lexical stem is usually used to refer to a heavy blow, e.g., with a fighting stick. In this case, thus, the lexical semantics of the lexical stem preempts the use with the classifier stem *slash(23)* and does not allow the combination with other classifier stems.

(4.22) ngu-nhi-bat-nu
    1SG.SLASH(23).FUT-2SG.DO-hit-FUT
    ‘I’ll hit you(sg).’ (Street 1989)

Finally, the use of *POKE(19)* with the lexical stem *thak* ‘dip’ as seen in (4.16) above is an example of a caused contact complex predicate in which the lexical stem can only combine with *POKE(19)*, and not *BASH(14)* or *SLASH(23)*. Here again, the semantics of the lexical stem as involving a dipping event determines that the classifier stem it should be used with is *POKE(19)*.

These fine-grained restrictions of the semantic concepts involved can be displayed in a hierarchy of selectional restrictions as illustrated in Figure 4.1. This hierarchy is an excerpt of the complex predicates of caused contact, in which also other classifier stems may be involved. For example, the lexical stem *lerrkperrk* ‘crush’ can combine with the classifier stems *HANDS(8)* and *FEET(7)* in addition to the classifier stems *BASH(14)*, *POKE(19)* and *SLASH(23)*, as can be seen in (4.23).
Figure 4.1: Hierarchy of selectional restrictions for complex predicates of caused contact

(4.23) a. \textit{ku tuntum mam-\textit{lerrkperrk}}
\textit{NC\textsubscript{anim} egg 1SG.S.HANDS(8).NFUT-CRUSH}
‘I crushed the egg in my hand.’

b. \textit{ngunungam-\textit{lerrkperrk}}
\textit{1SG.S.FEET(7).NFUT-crush}
‘I crushed the egg with my foot.’

c. \textit{bangam-\textit{lerrkperrk}}
\textit{1SG.SBASH(14).NFUT-crush}
‘I smashed it (with something heavy, like breaking a window with a rock; or like throwing a bottle onto the ground).’

d. \textit{dam-\textit{lerrkperrk}}
\textit{3SGS.POKE(19).NFUT-crush}
‘He cracked it (e.g., with the end of a spear).’

e. \textit{pan-\textit{lerrkperrk}}
\textit{1SGS.SLASH(23).NFUT-crush}
‘I smashed it (e.g., by hitting it with a stick).’

(Caudal et al. 2013a)

The lexical stem \textit{lerrkperrk} ‘crush’ can thus combine very productively with a whole range of classifier stems and would be high up in the hierarchy of...
118 An LCS blueprint analysis of complex predicates

selectional restrictions. The lexical stems *parrang* ‘numb’ and *yekum* ‘hit on ear’ can combine with the three classifier stems *bash*(14), *poke*(19) and *slash*(23) and are thus quite productive, too.

The finer-grained lexical semantic distinctions are found in the lower branches of the hierarchy. The lexical stems *rtal* ‘cut’ and *rlip* ‘hit’ can only combine with the classifier stems *bash*(14) and *slash*(23) while the lexical stem *thak* ‘dip’ can only combine with *poke*(19) etc.

The hierarchy of sectional restrictions thus makes fine-grained semantic concepts which play a role in the selection process of Murrinh-Patha classifier and lexical stem combinations visible. It provides an ordering into more productive complex predicates involving lexical stems that can combine with a whole range of different classifier stems and into less productive complex predicates in which more specific lexical stems are involved which can only combine with a small set or just one classifier stem.

### 4.2.2 Complex predicates of movement

Beyond the meaning of the classifier stems *bash*(14), *poke*(19) and *slash*(23) presented in the previous section, each of these classifier stems also has further meanings. All three classifier stems can be used in events involving movement. However, fine-grained meaning distinctions that restrict the combinatory possibilities of classifier stems and lexical stems in complex predicates of movement can be found for these combinations as well.

Barone-Nugent (2008) showed that *poke*(19) is used in linear movement, both in horizontal movement along the x-axis as in (4.24a) and in vertical movement along the y-axis such as in (4.24b).

(4.24)  
\[\text{pa-ngi-riwak-nu-ngintha}\]  
\[3\text{SgS.}\text{POKE(19).FUT-1SG.DO-follow-FUT-DU.F}\]  
‘They two will follow me.’  
(Fieldnotes R. Nordlinger)

\[\text{nga-wintigat-nu}\]  
\[1\text{SgS.}\text{POKE(19).FUT-descend-FUT}\]  
‘I’m going down.’  
(Fieldnotes R. Nordlinger)

In contrast, *bash*(14) and *slash*(23) can be found in the database with lexical stems of non-linear movement, such as circular or undirected movement as in (4.25a,b). They cannot be used with lexical stems of linear movement, as can be seen in (4.25c).

(4.25)  
\[\text{ba-rikat-nu}\]  
\[1\text{SgS.}\text{BASH(14).FUT-circuit-FUT}\]  
‘I’ll go around.’  
(Fieldnotes R. Nordlinger)
b. *ba-wintigat-nu
1SG.BASH(14).FUT-descend-FUT (Fieldnotes R. Nordlinger)

The classifier stem BASH(14) can also be used in other intransitive complex predicates of movement. In both cases in (4.26), it is not prototypically linear movement that is encoded in the complex predicates.

(4.26) a. nanthi ngurtinh bangampetuth
nanthi ngurtinh bangam-wetuth
NCres wave 3SG.BASH(14).NFUT-roll
‘The wave rolled.’ (Street 1989)

b. parnu bangampingkaparl
parnu bangam-wingkaparl
grass 3SG.BASH(14).NFUT-bent.over
‘The grass bent over.’ (Street 1989)

Another example involving undirected movement with the classifier stem SLASH(23) is presented in (4.27) which involves the lexical stem nangkartart ‘jump from one object to another’. This lexical stem can only be used with the classifier stem SLASH(23) and also denotes undirected, or at least non-linear, non-straight movement.

(4.27) ku mangki
NCanim monkey
pan-nangkartart=wurran
3SG.SLASH(23).NFUT-jump(RDP)=3SG.GO(6).NFUT
‘The monkey continually jumps from one (tree) to another.’ (Street 1989)

Additionally, combinations of lexical stems with SLASH(23) may involve the notion of an edge, which may be due to SLASH(23)’s basic meaning of the long side of an object. Thus, in (4.28), the movement happens along an edge.

(4.28) a. ngu-marda-nu
1SG.SLASH(23).FUT-move-FUT
‘I will move along the edge.’ (Street 1989)

b. ngu-rdertpart-nu
1SG.SLASH(23).FUT-skirt-FUT
‘I’ll skirt along the edge.’ (Street 1989)
An LCS blueprint analysis of complex predicates

The notion of an edge probably also accounts for the use of the classifier stem slash(23) in complex predicates as in (4.29). Here, slash(23) is used in combination with the lexical stems ringkarlpurl ‘moon rising’ and ngkamath ‘dawn coming’. In these cases the horizon is probably seen as an edge. (4.29b) can be seen as an extension of the concept of movement into the temporal domain.

(a) pu-ringkarlpurl-nu=purru
   3SGS.SLASH(23).FUT-rise-FUT=3SGS.GO(6).FUT
   ‘The moon will continually rise.’

(b) pu-ngkamath-nu
   3SGS.SLASH(23).FUT-come-FUT
   ‘Dawn will come.’

Thus, as with the caused contact use of the classifier stems bash(14), poke(19) and slash(23), a common meaning, i.e., that of movement, is evident in all three classifier stems. An overview over the system is presented in Figure 4.2. As with the caused contact complex predicates, movement complex predicates can also be formed with a range of other classifier stems, e.g., with go(6), feet(7) or hands(8), discussed in Chapter 3.

Specific meaning shades can be detected when the classifier stems are considered in detail. These meaning shades exclude certain combinations of classifier and lexical stems, for example lexical stems denoting non-linear movement with the classifier stem poke(19). For the non-linear movement complex predicates, a difference in behavior of the classifier stems bash(14)
and slash(23) can be observed as well. While both classifier stems can be used with the lexical stem rikat ‘circuit’, certain subtypes of non-linear movement exist which are formed with only one classifier stem.

4.2.3 Mouth-associated complex predicates

A third meaning range for the three classifier stems bash(14), poke(19) and slash(23) can be identified as “mouth-associated” actions. Barone-Nugent (2008) points out that poke(19) can be used in actions in which the mouth or mouth-associated body parts such as lips, teeth etc. play a role. He assumes that the perception of the tongue or the teeth as pointed ends may have enabled the use of the classifier stem poke(19) in actions in which the mouth plays a role.

The subgroup of mouth-associated actions is quite large; it comprises, among others, blowing, licking and chewing actions. Examples involving the classifier stem poke(19) are provided in (4.30).

(4.30) b. ku were
ku were
NC<sub>anim</sub> dog
parranigudhugudhuktha=pardi
parrani-gudhugudhuk-dha=pardi
3PLS.poke(19).pimpf-lick(4d)-pimpf=3PLS.be(4).pimpf
‘The dogs were continually licking him.’ (Street 1989)

c. mi nga-rlirr-nu
NC<sub>veg</sub> 1SGS.poke(19).fut-crunch.with.teeth-fut
‘I’ll crunch the food with my teeth.’ (Street 1989)

For the purpose of illustrating the different combinatory possibilities of the three classifier stems poke(19), bash(14) and slash(23), only two subgroups, speech actions and ingesting, are considered. The classifier stem poke(19) can be used both in complex predicates denoting speech actions and ingestion. Two examples involving the lexical stems dharrpu ‘ask’ and rluk ‘chew’ are provided in (4.31).

(4.31) a. nga-nhi-dharrpu-nu
1SGS.poke(19).fut-2SG.do-ask-fut
‘I’ll ask you.’ (Street 1989)

b. mi thawuy
mi thawuy
NC<sub>veg</sub> tobacco
damluk=wurran
dam-rluk=wurran
3SGS.poke(19).nfut-chew=3SGS.go(6).nfut
‘He chews tobacco.’ (Street 1989)
That poke(19) is indeed ambiguous between these meanings can be seen in combination with the lexical stem thap ‘try’. thap is a productive lexical stem which encodes the meaning ‘touch (to try out)’ with a range of other classifier stems. With the classifier stem poke(19), the resulting complex predicate is ambiguous between the reading ‘try out spoken words’ and ‘taste’, i.e., ‘taste’ should probably be understood as ‘touch with the mouth to try out’.

(4.32) a. murrinh mani nga-thap-nu
   NC MOD 1SG.POEKE(19).FUT-try-FUT
   ‘I’ll test out the spoken words.’ (Street 1989)

b. kura parranthap
   kura parram-thap
   NC_water 3PLS.POEKE(19).NFUT-try
   ‘They tasted the water.’ (Street 1989)

The classifier stem poke(19) can thus be used both in complex predicates of speech actions and of ingestion. In contrast, the classifier stem bash(14) is not used in complex predicates denoting speech actions. It can be used in complex predicates of ingesting, such as in (4.33). The lexical stem gatkat ‘eat until satisfied’ combines with bash(14) in (4.33a). In (4.33b), bash(14) combines with thap ‘touch, try out’ and only has the meaning ‘taste’.

(4.33) a. ku ngen ba-gatkat-nu
    NC_anim meat 1SG.BASH(14).FUT-eat.until.satisfied
    ‘I’ll eat meat until I’m satisfied.’ (Street 1989)

b. mi kanhi ba-thap-nu
    NC_veg DEM 1SG.BASH(14).FUT-try
    ‘I’ll taste this food.’ (Street 1989)

slash(23) behaves the other way around, i.e., it can be readily used in speech actions, but not in complex predicates of ingesting. The use of slash(23) in speech actions can be seen in (4.34) which involves the lexical stems rerda ‘blame’ and marda ‘request’.

(4.34) a. pan-ngi-rerda=kanam
    3SGS.SLASH(23).NFUT-1SG.DO-blame=3SGS.BE(4).NFUT
    ‘He continually blames me.’ (Street 1989)

b. ngu-nhi-marda-nu mani?
    1SGS.SLASH(23).FUT-2SG.DO-request-FUT MOD
    ‘Can I request of you?’ (Street 1989)
SLASH(23) is usually not used in complex predicates of ingesting, according to the database. The only two examples which can be found in Street (1989) of SLASH(23) being used in a complex predicate of ingesting are given in (4.35). In these examples it can probably be argued that it is not the ingesting that is important for the selection of the classifier stem but rather the action that leads to the food being brought close to or into the mouth.

(4.35)  

a. ku lapi pan-purl  
\texttt{N}_{\text{Canim}} \text{ rib membrane} \quad \texttt{3SG} \texttt{.SLASH(23).NFUT}  
‘He dragged the membrane from the rib bone with his teeth.’  
\textit{(Street 1989)}

b. ku ngalek  
k\texttt{u ngalek}  
\texttt{N}_{\text{Canim}} \text{ mosquito}  
\texttt{puninkatattha=dini} \texttt{punin-nkatat-dha=dini} \texttt{3SG} \texttt{.SLASH(23).PIMPFF-catch-PIMPFF=1SG.SIT(1).PIMPFF}  
‘He was catching mosquitoes (with his tongue).’  
\textit{(Street 1989)}

Thus, also for complex predicates of mouth-associated actions, a common domain of meaning can be identified with all three classifier stems. An overview over the system is presented in Figure 4.3.
4.2.4 Multiple dimensions for classifier stems

The previous sections discussed various types of complex predicates in which the classifier stems BASH(14), POKE(19) and SLASH(23) can be used. It could be shown that the classifier stems pose restrictions on the lexical stems that they combine with. However, the meaning that the classifier stems themselves carry has not become completely clear. Based on the discussed combinations, one could assume that the classifier stems are very bleached of any lexical semantic content and that it is the lexical stem that contributes most information. In this view, which classifier stem combines with which lexical stem would be pure idiosyncratic coincidence. However, there is evidence that the classifier stems BASH(14), POKE(19) and SLASH(23) indeed are ambiguous and have these various meaning dimensions. This can be seen in certain contexts in which the classifier stems themselves can be used to denote these meaning dimensions.

Certain lexical stems such as wirntay ‘miss’, ngkamat ‘miss’ and rikerdek ‘finish’ foreground the meaning of the classifier stem. These combinations can thus be used to study what meaning range can be covered by the classifier stems BASH(14), POKE(19) and SLASH(23) themselves.

Examples of the classifier stems denoting specifically shaped instruments are provided in (4.36) in which the lexical stem wirntay ‘miss’ is used with all three classifier stems. According to Street (1989), wirntay ‘miss’ in combination with POKE(19) means ‘miss with a spear’, while it means ‘miss with a stone or short spear’ with the classifier stem BASH(14) and ‘miss with a stick’ with the classifier stem SLASH(23).

\[
\text{(4.36) a. nga-wirntay-nu} \\
1SGS.POKE(19).FUT-miss-FUT \\
‘I will miss (with a spear).’ (Street 1989)
\]

\[
\text{b. bangam-na-wirntay} \\
1SGS.BASH(14).NFUT-3SG.BEN.M-miss \\
‘I missed him (with a stone).’ (Street 1989)
\]

\[
\text{c. thu thay pan-na-wirntay} \\
\text{NC\_weapon stick 1SGS.SLASH(23).NFUT-3SG.BEN.M-miss} \\
‘I missed (hitting) him with the stick.’ (Street 1989)
\]

For the classifier stem POKE(19), this can also be seen in (4.37) with the lexical stem ngkatmut ‘miss’. It seems that POKE(19) here denotes the action of writing which can be deduced from actions that are carried out with the tip of a long object as well.

\[
\text{(4.37) murrinh pana ngamkatmut=ngurran} \\
murrinh pana ngam-ngkatmut=ngurran \\
\text{NC\_speech DEM 1SGS.POKE(19).NFUT-miss=1SGS.GO(6).NFUT} \\
‘I continually miss writing down that word.’ (Street 1989)
\]
(4.36) and (4.37) so illustrate the use of the classifier stems of caused contact with an instrument. However, according to Street (1989), the combination of the lexical stem *wirntay* ‘miss’ with the classifier stems POKE(19) and SLASH(23) can be interpreted as ‘miss giving a message’ as well. This shows that POKE(19) and SLASH(23) can denote speech actions even without lexical stems of speech.

Similar observations can also be made for the combination of the classifier stems with the lexical stem *rikerdek* ‘finish’. Street (1989) states that POKE(19) can mean ‘finish writing’ if combined with *rikerdek*. Additionally, it can also mean ‘finish eating’, as can be seen in (4.38a). The classifier stem BASH(14) in combination with *rikerdek* is given as ‘finish drinking’ by Street (1989), as can be seen in (4.38b).

(4.38) a. *mi* parrantikerdek
    mi parram-rikerdek
    NC*veg* 3PLS.POKE(19).NFUT-finish
    ‘They(pl) finished off the food.’ (Street 1989)

b. *kura* warra ngadha da-rikerdek-nu
    NC*water* first 2SGS.BASH(14).FUT-finish-FUT
    ‘You(sg) finish your drink first.’ (Street 1989)

Thus, the same patterns that were observed in the previous sections can be observed with these lexical stems. While the classifier stem POKE(19) is used in mouth-associated actions of ingesting and speech, the classifier stem BASH(14) is only used in actions involving ingestion and SLASH(23) is only used in actions involving speech. The fact that these differences evolve even with these neutral lexical stems that do not bring with themselves a specific meaning from this semantic range provides evidence that the classifier stems indeed express these different meaning dimensions.

The data thus shows that the same classifier stems can be used in a wide variety of different complex predicates. For some complex predicates, it is quite obvious why a specific classifier stem is used, e.g., in the examples involving caused contact above. Similarly, even for quite different complex predicates such as the caused contact and movement complex predicates involving POKE(19), it is understandable why the same classifier stem can be used. The linear movement is so prominent in the meaning of the prototypical use of the classifier stem as involving poking actions that the classifier stem can be used even in contexts in which no external source controls the movement.

In contrast, for some complex predicates it is not clear why the same classifier stem is used. For example, Barone-Nugent (2008) suggests that POKE(19) is licensed in complex predicates in which the mouth plays a role because the teeth and the tongue can be perceived as the pointed end
of a long object. However, this extension is quite difficult to accommodate, especially because the explanation does not extend to the cases in which \texttt{slash(23)} is used in speech actions. If \texttt{poke(19)} is used in mouth-associated actions because the teeth and the tongue are perceived as pointed ends of long objects, it is not clear why at the same time the side of a long object should play a role in speech actions. Nordlinger (2012a) also states that the reading of \texttt{poke(19)} denoting speech actions is as salient as its reading involving a long pointed object. This also speaks in favor of assuming undervived meanings between these two uses.

It is thus questionable whether all combinatory possibilities rely on cognitive, perceptual factors. In some cases, it may be pure morphological coincidences that the same classifier stem is used. Barone-Nugent (2008) argues for such an explanation for the classifier stem \texttt{hands(8)} which is used in actions performed by the hands but which is also used in speech acts. He points out that the paradigm for the classifier stem \texttt{hands(8)} is very similar to the paradigm of the classifier stem \texttt{say/do(34)} and that this similarity together with the similarity in meaning for the non-speech acts may have licensed the use of \texttt{hands(8)} with speech acts.

Nordlinger (2012a) thus assumes that the Murrinh-Patha complex predication system cannot be explained in purely semantic terms but that a range of formal considerations such as morphological coincidences play a role, too. The approach of various dimensions of the hierarchies of selectional restrictions advocated for here aims at looking at the combinatory possibilities and focuses on the factors which play a role in the selection of the classifier stems (such as caused contact, movement, etc.) and the subtypes which are determined by different classifier stems (linear vs. non-linear movement etc.). It does not assume a necessary cognitive semantic link between the different dimensions. The links may be cognitively determined, such as in the case of \texttt{poke(19)} being used in complex predicates of linear movement, but they may also be pure morphological coincidences, such as the case of \texttt{hands(8)} being used in complex predicates of speech and caused contact with hands.

To sum up, the hierarchies of selectional restrictions model the fact that classifier stems have a rather general meaning which allows them to combine with a wide range of different lexical stems. In this system, classifier stems can have different meaning dimensions while lexical stems usually have only one meaning dimension. The hierarchies so sort the different complex predicates into various groups and subgroups. As such, the discussion of the combinatory possibilities and the ordering into hierarchies of selectional restrictions is a relation-based approach to lexical semantics.

This relation-based approach, however, does not provide a decomposition of the meaning of the complex predicate as primitive-based approaches such as Jackendoff’s (1990) LCSs can offer. To model the selectional restrictions that the classifier stem poses on the lexical stem, however, a decomposition
of the semantics of the complex predicate into meaning aspects provided by the classifier stem and meaning aspects provided by the lexical stem is useful. The following section argues for such an approach, making use of LCS blueprints to model the meaning decomposition of Murrinh-Patha complex predicates.

4.3 An LCS blueprint analysis

The previous section provided hierarchies of selectional restrictions for the classifier stems BASH(14), POKE(19) and SLASH(23) which showed that fine-grained semantic distinctions restrict the combinatory possibilities of Murrinh-Patha classifier and lexical stems, and not their argument structure. This section is now concerned with deriving the argument structure from the hierarchies of selectional restrictions.

For the purpose of deriving the argument structure from the hierarchies of selectional restrictions, LCS blueprints are used. Section 4.3.1 argues that LCS provide the right granularity of information to model the selectional restrictions. Section 4.3.2 then shows that organizing the LCS into blueprints ensures an effective way of modeling the restrictions. Finally, Section 4.3.3 discusses how the LCS blueprints can be linked to the hierarchies of selectional restrictions in detail.

4.3.1 Motivation for using LCS

Traditional accounts of argument selection assume thematic roles to model the requirements that the verb sets for its arguments. For example, different thematic role requirements are used to explain the different behavior of intransitive verbs in a range of cross-linguistic tests (see e.g., Ahmed 2010 for discussion) for the unaccusative/unergative distinction that was proposed by Perlmutter (1978).

However, other criteria may play a role in the restrictions that the verb sets for its arguments as well. Ahmed (2010), for example, questions the function of thematic roles for the selection criteria for Urdu intransitive verbs. He bases his claim on the contrast in (4.39) in which the same intransitive verb ur ‘fly’ is used. The verb would be considered to be an unergative verb. However, depending on the animacy of the subject, the verb behaves differently under a range of tests proposed by Bhatt (2003) for the unergative/unaccusative distinction in Urdu. One such test is the reduced relative construction, which is ungrammatical for unergative verbs but grammatical for unaccusative verbs. As Ahmed (2010) shows, the verb ur ‘fly’ behaves differently in this test. The reduced relative construction is ungrammatical if the subject is animate as in (4.39c) while it is grammatical if the subject is inanimate as in (4.39d).
An LCS blueprint analysis of complex predicates

(4.39) a. ciryaa  ur-ii
    bird.FSG  fly-PFV.FSG
    ‘The bird flew.’ (Urdu, Ahmed 2010:8)

b. patang  ur-ii
    kite.FSG  fly-PFV.FSG
    ‘The kite flew.’ (Urdu, Ahmed 2010:8)

c. *ur-ii (huu-ii) ciryaa
    fly-PFV.FSG  be-PFV.FSG  bird.FSG
    ‘The flown bird’ (reduced relative test) (Urdu, Ahmed 2010:8)

d. ur-ii (huu-ii) patang
    fly-PFV.FSG  be-PFV.FSG  kite.FSG
    ‘The flown kite’ (reduced relative test) (Urdu, Ahmed 2010:9)

Thus, semantic criteria such as animacy may play a role in the selection process of verbs which can be tested in different constructions. Jackendoff (1990) even assumes more information to be relevant for selectional restrictions. For example, Jackendoff (1990:164f) proposes the LCS in (4.40) for the verb butter. He argues that BUTTER can be an implicit argument or a selectional restriction on the theme object. He bases his argument on the anomaly of sentences such as (4.41) which shows that BUTTER puts a selectional restriction on the object of with, namely that it should be a buttery substance.

(4.40) \[
  \text{CAUSE ([[Thing}]}, \text{BECOME ([BE ([[BUTTER]], [ON ([[Thing]])]])})\]

(4.41) #We buttered the bread with pineapple juice.

Restrictions of the same granularity can be found in Murrinh-Patha classifier and lexical stem combinations, as was discussed in the previous section. For example, the lexical stem mel ‘flatten’ can combine with the classifier stems HANDS(8), BASH(14) and SLASH(23), but not with the classifier stem POKE(19), because flattening an object with the tip of a long pointed object is not possible (or at least very unusual). It is thus instruments (denoted by the classifier stem) that pose restrictions on the result state (denoted by the lexical stem) of Murrinh-Patha complex predicates.

To see how this can be implemented with LCSs, a brief discussion of how instruments are modeled within LCSs is needed. Jackendoff (1990:142f) proposes that instruments should be modeled with a BY function which turns events into means modifiers. For example, he proposes the LCS in (4.42b) for the sentence in (4.42a). It presents the standard LCS for a causative verb assumed by Jackendoff (1990), combined with the BY function that
is introduced by the instrument. The by function then embeds a cause
relation in which Phil acts on the key so that the key can act on the door.
In this way, the instrument is seen as an intermediary for the actor’s action.

(4.42) a. Phil opened the door with a key.

\[
\begin{align*}
&\text{CAUSE([PHIL], \text{BECOME BE ([DOOR], [OPEN])})]} \\
\text{AFF}^{-}([\text{PHIL}, [\text{DOOR}]})] \\
\text{BY} &\left[\text{CAUSE([PHIL], \text{AFF}^{-}([\text{KEY}, [\text{DOOR}]})]) \right]
\end{align*}
\]

This modeling can now be transferred to Murrinh-Patha complex predicates.

(4.43) provides an example of the combination of mel ‘flatten’ with the
classifier stem hand(8). As can be seen, the classifier stem contributes an
instrument (by hand) to the meaning of the complex predicate.

(4.43) mam-mel

\[
\begin{align*}
\text{1SGS.HANDS(8).N.FUT-flatten} \\
\text{‘I flattened it out by hand’ (Street 1989)}
\end{align*}
\]

Applying Jackendoff’s (1990) LCS for instruments to this example yields the
LCS in (4.44). HAND fulfills the role of the instrument, FLAT is the result
state in this case. The subjects of the cause relations are coindexed by α,
forming the subject of the complex predicate. Similarly, the argument of the
AFT\(^{-}\) function and the argument of the BE function are coindexed (marked
by β), forming the object of the complex predicate. This then reflects the
reading that someone causes something to be flat by acting on it with his or
her hand. Similarly to the BUTTER example above, one can assume here
that the instrument, HAND in this case, poses a restriction on which result
states are possible.

(4.44) HANDS(8)-mel ‘flatten by hand’

\[
\begin{align*}
\text{CAUSE ([\text{THING}[\alpha], \text{BECOME ([BE ([THING][\beta], [FLAT]])})])} \\
\text{BY} &\left[\text{CAUSE([THING][\alpha], \text{AFF} ([\text{HAND}, [THING][\beta]])]) \right]
\end{align*}
\]

The LCSs thus mirror the constraints that were modeled by the hierarchies of
selectional restrictions for the complex predicates involving caused contact.
As is discussed in Section 4.3.3, the distinction between manner of motion

\^2Jackendoff (1990) actually makes a distinction between different cause functions
which are not important for the discussion at hand, so that the formula has been slightly
simplified. The relation AFT\(^{-}\) is part of the Action Tier and indicates that the second
argument is negatively affected.

\^3The Action Tier has been left out as it is not discussed in the remainder of the
examples.
that does not express a path, modeled by Jackendoff (1990) with the function \textit{MOVE}, and motion verbs denoting a path, modeled by Jackendoff (1990) with the function \textit{GO}, also reflects the selectional restrictions of Murrinh-Patha complex predicates to a large extent.

(4.44) illustrates a further advantage of modeling the argument structure with LCSs, namely that LCSs serve to make the decomposition of the meaning of the complex predicate visible. This is especially useful for Murrinh-Patha complex predicates as classifier and lexical stems contribute different meaning aspects to the LCS of the complex predicate, at least in productive combinations. For example, in (4.44), the classifier stem contributes the instrument \textit{HAND} while the lexical stem contributes the result state \textit{FLAT}. Due to the decompositional approach of LCSs, LCSs can make these contributions visible, in contrast to relation based approaches. This is discussed in further detail in the following section.

To sum up, LCSs have been chosen because they provide the right granularity of information to incorporate the most important concepts that are relevant in the selectional restrictions of Murrinh-Patha complex predicates. Murrinh-Patha is typologically extraordinary in having grammaticalized a sibling relation in its number system. Similarly, Murrinh-Patha is typologically special in that concepts such as different instrument shapes play a role in the combinatory possibilities of classifier and lexical stems. That is, the concepts of the specifically shaped instruments (hands, feet, flat solid objects for the classifier stem \textit{BASH(14)} etc.) are grammaticalized in Murrinh-Patha classifier stems. Murrinh-Patha complex predicates are thus different from the previously described complex predicates as they can be considered semantic complex predicates.

4.3.2 Blueprints for classifier and lexical stem combinations

The previous section suggested the use of LCSs to model the argument structure of complex predicates in Murrinh-Patha. However, it was shown in Section 4.1 that existing LCS approaches to complex predicate formation do not yield elegant and satisfying results for Murrinh-Patha. Murrinh-Patha complex predicates exhibit a great variety in their combinatory possibilities, due to the fact that it is not the argument structure but various semantic concepts that account for the combinatory possibilities. An account along the lines of previous approaches would thus either involve a large set of different rules or a range of different LCSs for the same classifier stem. However, such an approach then loses explanatory power, as the rules are not restricted and so overgenerate.

A purely enumerative approach, in which each possible classifier and lexical stem is listed, is not satisfying either. Across the Murrinh-Patha data, groups of classifier and lexical stem combinations can be determined which behave alike. For example, the anticausative, resultative construction with
the classifier stem sit(1), discussed in Chapter 3, not only occurs with the lexical stem lerrkperrk ‘crush’, but with a whole range of change of state lexical stems such as e.g., parl ‘break’, warnuta ‘split open’ or buwurr ‘fracture’. Similarly, the lexical stem lerrkperrk ‘crush’ can also combine with a range of different classifier stems, e.g., with the classifier stems FEET(7), HANDS(8), BASH(14), POKE(19) and SLASH(23), as was illustrated in (4.23).

A separate lexical entry for each of these combinations would thus be very inefficient.

For this reason, this section proposes to model the Murrinh-Patha data with LCS blueprints. The idea of blueprints is influenced by the notion of templates as used in the grammar development platform xle (Crouch et al. 2015). Templates are used in the implementation to model regularities that can be found in the lexical entries (Dalrymple et al. 2004). However, they can also be useful in theoretical considerations, as was suggested by Asudeh et al. (2008) in their treatment of constructions (in the Construction Grammar sense (Fillmore 1988, Goldberg 1995)) in lfg. Similarly, Bouma & Kuhn (2009) use templates in their treatment of the Dutch laten-causative as a complex predicate with restrictions on the combinatory possibilities.

In the approach proposed here, the idea of templates is combined with the modeling of the argument structure with LCSs. These LCS blueprints are defined for classes of different classifier and lexical stem combinations in Murrinh-Patha. What this means is that the rule of how a classifier and lexical stem combines is bound to the combination itself. This contrasts with the analyses of Urdu by Butt (1995) or of Wagiman by Wilson (1999) in which the rules are bound to the light verb only, accounting for the combinations of this light verb with all elements that take part in this complex predicate formation.

The LCS blueprints specify which information is contributed by the classifier stem and which information is contributed by the lexical stem. The basic underlying ideas are illustrated with the behavior of the change of state lexical stems such as lerrkperrk. Besides their use in anticausative/resultative constructions with sit(1) (repeated in (4.45a)), they can also be used in transitive causative complex predicates, for example with the classifier stems HANDS(8) ((4.45b)) and FEET(7) ((4.45c)).

(4.45) a. dim-lerrkperrk  
   3SGS.SIT(1).NFUT-crush
   ‘It’s smashed.’ (Fieldnotes R. Nordlinger)

b. ku tumtum mam-lerrkperrk  
   NC_anim egg 1SGS.HANDS(8).NFUT-crush
   ‘I crushed the egg in my hand.’ (Fieldnotes R. Nordlinger)

c. ngunungam-lerrkperrk  
   1SGS.FEET(7).NFUT-crush
   ‘I crushed the egg with my foot.’ (Fieldnotes R. Nordlinger)
For these combinations, different blueprint LCSs are needed to model the different behavior in (4.45a) versus (4.45b,c). The blueprint LCS for the causative complex predicate is defined as in (4.46). It states that the complex predicate expresses the meaning that something or someone ($\alpha$) causes something ($\beta$) to become a certain result state with the help of some specific instrument. In this blueprint, the result state is provided by the lexical stem while the instrument is provided by the classifier stem. For example, the lexical stem *lerrkperrk* 'crush' contributes the result state CRUSHED, while the classifier stems HANDS(8) and FEET(7) contribute the instruments HAND and FOOT respectively.  

\[ \text{(4.46) CLASSIFIER STEM + lexical stem denoting change of state} \]

\[
\begin{align*}
&\text{CAUSE ([[\text{Thing} \alpha]], \text{BECOME} ([\text{BE ([[\text{Thing} \beta]], [\text{RESULT}]])])))} \\
&[\text{BY [CAUSE ([[\text{Thing} \alpha]], [\text{AFF ([INSTRUMENT], [[\text{Thing} \beta]]])])}]}
\end{align*}
\]

In this view, the classifier stem and the lexical stem do not bring with themselves a complete LCS, but just a specifically shaped instrument specified by the classifier stem or a specific result state specified by the lexical stem. The lexical entries of the classifier stem and the lexical stem thus only consist of this information, as is illustrated in (4.47).

\[ \text{(4.47) } \begin{align*}
\text{HANDS(8):} & \quad \text{INSTRUMENT = HAND} \\
\text{FEET(7):} & \quad \text{INSTRUMENT = FOOT} \\
\text{lerrkperrk:} & \quad \text{RESULT = CRUSHED}
\end{align*} \]

The basic structure of the LCS, e.g., that (4.46) involves a causative, transitive LCS, is determined by the combination of the classifier and lexical stem. Each combination specifies which blueprint LCS it uses. This is further discussed in the following section that links the LCS blueprints to the hierarchies of selectional restrictions.

Because it is the classifier and lexical stem together that specify which blueprint LCS is used, they determine the transitivity of the complex predicate together. This captures the view advocated for in Chapter 3 that argument structure is not determined by the classifier stem, as so-called transitive classifier stems can occur in intransitive complex predicates and vice versa. It also solves the problem that assuming an argument structure for lexical stems is difficult as they never function as the sole verbal predicate. That is, while an approach to complex predicate formation that assumes a full LCS

---

\(^{4}\text{As in Jackendoff's (1990, 164f) examples "\#We buttered the bread with pineapple juice" vs. "We buttered the bread with cheap margarine", the instrument denoted by the classifier stem can be a selectional restriction on the instrument. That means that it is possible to include overt instruments which then replace the instrument denoted by the classifier stem in the LCS, but the overt instrument has to be compatible with the instrument denoted by the classifier stem.}\)
for the lexical stem has to decide whether lerrkperrk ‘crush’, for example, is intransitive or transitive, the approach proposed here only has to specify that lerrkperrk ‘crush’ contributes the result state CRUSHED.

Consider now the blueprint LCS in (4.48) for anticausative/resultative complex predicates with sit(1) in (4.45a). The resulting complex predicate is a stative complex predicate. As in (4.46) above, the lexical stem contributes the result state. The classifier stem does not overtly contribute a feature in the LCS, but the use of the blueprint LCS is restricted to the classifier stem sit(1), i.e., the blueprint LCS is only called by the combination of sit with a lexical stem denoting change of state.

\[(4.48) \text{sit}(1) \text{-lexical stem denoting change of state} \]

\[\text{BE } ([\text{Thing } A], [\text{RESULT}])\]

Assuming no argument structure for the lexical stem thus solves the problems that other approaches have with this construction. Because the approach does not assume that lerrkperrk ‘crush’ is transitive, it does not have to deal with violations of monotonicity.

The LCS blueprint for causative complex predicates of change of state displayed in (4.46) accounts for a large range of classifier and lexical stem combinations as it allows the combination of different classifier stems with different lexical stems. This LCS blueprint thus accounts for many different productive complex predicates. In contrast, the LCS blueprint of anticausative/resultative complex predicates in (4.48) only accounts for complex predicates involving the classifier stem sit(1). It is thus more restricted, but still semi-productive in the sense that it predicts that a whole range of change of state lexical stems can form this pattern.

The LCS blueprints can thus be used for very productive combinations of classifier and lexical stems as well as for smaller subgroups of lexical stems that can combine with a small set of classifier stems or with only one classifier stem. As the approach incorporates the idea of more and less productive combinations, it also extends naturally to opaque, lexicalized combinations of classifier and lexical stem combinations. An example of an opaque combination involving the classifier stem HANDS(8) is provided in (4.49) in which case it is not clear why the classifier stem HANDS(8) is used.

\[(4.49) \text{mam-pun-mardaraki} \]

\[3SGS.HANDS(8).\text{NFUT-3PL.DO-disappoint}\]

‘He disappointed them(pl).’ (Street 1989)

The LCS for this combination is provided in (4.50). As can be seen, the LCS is very similar to the LCS blueprint for causative complex predicates presented in (4.46), with the exception that no instrument is specified in this case. That is, the classifier stem HANDS(8) is very bleached in this combination,
which can be seen by the fact that it does not contribute an instrument to the LCS.

(4.50) hands(8) + mardaraki ‘disappoint’

\[
\text{CAUSE } ((\text{Thing} | \text{A}, \text{BECOME } ([\text{BE } ([\text{Thing} | \text{A}, \text{DISAPPOINTED}]))]))
\]

However, even for these lexicalized combinations, an LCS blueprint can be defined, as many lexicalized combinations involve causative complex predicates without specifying the instrument involved. For example, the lexical stem riwiye ‘pollute’ combines with poke(19) as can be seen in (4.51a). For this lexicalized combination, the LCS is provided in (4.51b).

(4.51) a. kura nga-riwiye-nu

\[\text{NCwater 1SG.POKE(19).FUT-pollute-FUT}\]

‘I will pollute the water.’

(Street 1989)

b. POKE(19) + riwiye ‘pollute’

\[
\text{CAUSE } ((\text{Thing} | \text{A}, \text{BECOME } ([\text{BE } ([\text{Thing} | \text{A}, \text{POLLUTED}]))]))
\]

As can be seen from the contrast in (4.50) and (4.51b), the LCSs are just distinguished through their result state. An LCS blueprint can thus be proposed for both combinations as in (4.52) which accounts for all causative complex predicates which do not specify a specific instrument.

(4.52) classifier stem + lexical stem denoting change of state

\[
\text{CAUSE } ((\text{Thing} | \text{A}, \text{BECOME } ([\text{BE } ([\text{Thing} | \text{A}, \text{RESULT}]))]))
\]

This LCS blueprint is called by the combination of hands(8) with the lexical stem mardaraki ‘disappoint’ for which the lexical stem specifies that the result state is DISAPPOINTED and it can be called by the combination of the classifier stem POKE(19) with the lexical stem riwiye ‘pollute’ which specifies that the result state is POLLUTED. Thus, even for lexicalized combinations, the LCS blueprint approach has advantages over a purely enumerative approach as the same LCS blueprints can be used for different lexicalized combinations. Moreover, the LCS blueprints make obvious that what is bleached for these combinations is that the classifier stems do not specify specific instruments.

To sum up, the LCS blueprint approach assumes that two separate lexical entries are important for complex predicate formation. On the one hand, speakers store the information the classifier and lexical stems encode, e.g., that classifier stems may contribute a certain instrument shape or that lexical stems contribute a certain result state to the complex predicate. On the other hand, speakers also store how a certain classifier stem can combine with a certain lexical stem, i.e., which LCS blueprint the combination uses.
An LCS blueprint analysis of complex predicates 135

\[ \text{HANDS}(8) + \text{lerrkperkk} \text{ ‘crush’ :} \]

\[
\begin{align*}
\text{CAUSE} ( & \text{[Thing} \ [\bar{A}], \ \text{BECOME (\text{BE (}[\text{Thing} \ [\bar{A}], \ \text{CRUSHED}]))]} ) \\
\text{BY} & \ \text{[CAUSE([Thing} \ [\bar{A}], \ \text{AFF} (\text{HAND}, [\text{Thing} \ [\bar{A}])))]} \\
\text{SUBJ} & \\
\text{OBJ} & \\
\text{PRED} & \text{‘HANDS}(8) + \text{lerrkperkk <([\text{SUBJ}],[\text{OBJ}])’}
\end{align*}
\]

Figure 4.4: An example of the linking between LCS and f-structure

In this way, the approach can also explain how unknown combinations may be accommodated. If an unknown combination is uttered, the hearer may try to insert the lexical information of the classifier and lexical stems into a productive LCS blueprint. If the combination then renders a feasible action that meets the selectional restrictions, e.g., in that the instrument fits to the result state, the classifier and lexical stem combination can be understood.

The insertion of the classifier and lexical stem information into the LCS blueprints happens in the lexicon. Once the insertion is successful, an LCS of the complete complex predicate is the result. This LCS can then be linked to syntax, for example via the mapping formalism proposed by Butt (1995) from LCS to f-structure, who proposes a set of rules how the arguments of the LCS can be mapped onto the grammatical relations in f-structure. An example for the complex predicate formed with the classifier stem HANDS(8) and the lexical stem lerrkperkk is provided in Figure 4.4 in which the agent is mapped onto the subject and the theme is mapped onto the object.\(^5\)

4.3.3 LCS blueprints and hierarchies of selectional restrictions

The previous section laid out the main ideas of the LCS blueprint analysis, namely that classifier and lexical stems do not contribute whole LCSs that need to be compatible but that they only contribute certain meaning aspects which are then inserted into an LCS blueprint. Which LCS blueprint is used is determined by the classifier and lexical stem together, whose compatibility is modeled by the hierarchies of selectional restrictions presented in Section

\(^5\)Butt (1995) requires her formalism to operate on the syntactic level, but emphasizes that it can also be used for morphological complex predicates like the Urdu causative.
Figure 4.5: Hierarchy of selectional restrictions for complex predicates of caused contact with links to LCS blueprints

4.2. This section now lays out how the LCS blueprints are linked to the hierarchies of selectional restrictions and how they interact.

For the purpose of modeling the linking between LCS blueprints and hierarchies of selectional restrictions, names are introduced for the LCS blueprints. These names consist of the @-symbol as a marker for LCS blueprints and an abbreviation, for example cos for change of state. The LCS blueprint for complex predicates denoting change of state is thus marked with @cos, which can be seen in (4.53).

\[(4.53) \text{@cos: classifier stem + lexical stem denoting change of state} \]
\[\text{CAUSE } ([\text{Thing}]_A^{\alpha}, \text{BECOME } ([\text{BE } ([\text{Thing}]_A^{\beta}, \text{RESULT}])[)])] \]
\[\text{BY } \text{CAUSE } ([\text{Thing}]_A^{\alpha}, [\text{AFF } ([\text{INSTRUMENT}], [\text{Thing}]_A^{\beta}))])] \]

These abbreviations are then also used in the hierarchies of selectional restrictions to specify which LCS blueprint is used by a specific classifier and lexical stem combination. An example for complex predicates of caused contact is provided in Figure 4.5. As can be seen, for example, the combinations of the classifier stems BASH(14), SLASH(23) and POKE(19) with the lexical stem parrang ‘numb’ all use the LCS blueprint @cos. The hierarchies of selectional restrictions are thus a hierarchical ordering of the lexicon entries for the combinations of classifier and lexical stems.
As has been discussed in Section 4.2, the hierarchies of selectional restrictions model the fine-grained differences between the classifier and lexical stem combinations. For example, for the caused contact complex predicates, the hierarchy showed that the shape of the instrument may restrict the possibilities of combinations with a specific lexical stem. That is, the result state specified by the lexical stem and the instrument need to be compatible, e.g., the classifier stem POKE(19) is not used with the lexical stem *mel* ‘flatten’ because flattening something with the tip of a long pointed object is probably impossible or at least very uncommon. It is thus semantic compatibility that is modeled with the hierarchy of selectional restrictions.

These selectional restrictions do not have a one to one correspondence to LCS blueprints, as combinations occupying the same node in the selectional restrictions may have a different LCS. This is discussed in the following for the case study of complex predicates of caused contact with the classifier stems BASH(14), SLASH(23) and POKE(19).

As can be seen in Figure 4.5, the lexical stems *mel* ‘flatten’ and *rlip* ‘hit’ can both combine with the classifier stems BASH(14) and SLASH(23). Examples with the classifier stem SLASH(23) are repeated in (4.54).

1. *angu-mel-nu*
   
   1SGS.SLASH(23).FUT-flatten-FUT
   
   ‘I will flatten it out (with a stick/pipe).’  
   (Street 1989)

2. *panperlip*
   
   pan-we-rlip
   
   3SGS.SLASH(23).NFUT-IPB-hit
   
   ‘He hit him on the head (with a stick).’  
   (Street 1989)

While these combinations thus occupy the same node in the hierarchy of selectional restrictions, they do not make use of the same LCS blueprint. The two causative complex predicates are examples of the well-known distinction between hit verbs and break verbs (Fillmore 1970). The complex predicate with the lexical stem *mel* ‘flatten’ belongs to the class of break verbs, i.e., it forms a complex predicate of change of state and consequently uses the LCS blueprint @cos.

In contrast, complex predicates with the lexical stem *rlip* ‘hit’ do not form change of state complex predicates and therefore do not call the cos blueprint. Instead, they form complex predicates of hitting, marked with “@hit” in the hierarchy of selectional restrictions.

In contrast to complex predicates of breaking, in complex predicates of hitting, the action does not have to have an effect on the object. The result state, in this case, is that the instrument, denoted by the classifier stem, is in contact with the object. (4.55) presents the LCS blueprint for this type of complex predicate formed with lexical stems of hitting, which follows the LCS of verbs of hitting proposed by Jackendoff (1990:143). As above, the
classifier stem fills in the information for the instrument. In contrast to above, however, the lexical stem does not contribute directly to the LCS, as the contribution of the various lexical stems, e.g. whether the hit was heavy (meaning of dheng) or only once (meaning of the lexical stem bat), is not part of conceptual structure, but a finer-grained semantic distinction.

(4.55) $\@$HIT: CLASSIFIER STEM + lexical stem of hitting

$$\left[ \text{CAUSE} \left( \text{[Thing]} A^\alpha, \right. \left. \text{BECOME} \left( \text{[INSTR]}, \text{[AT} \left( \text{[Thing]} A^\beta \right) \text{]} \right) \right) \left[ \text{BY} \left[ \text{CAUSE} \left( \text{[Thing]} A^\alpha, \text{[AFF} \left( \text{[INSTR]}, \text{[Thing]} A^\beta \right) \text{]} \right) \right] \right] \right]$$

Finally, the lexical stem yekum ‘hit on ear’, illustrated in (4.18) and repeated in (4.56), is special in this range of lexical stems of hitting as it contributes to the LCS of the complex predicate the place of contact, i.e., the ear. The lexical stem yekum is probably a lexicalized form of the combination of the incorporated body part ye ‘ear’ and a lexical stem kum. The difference between the lexical stems in (4.56) and (4.54b) is that rlip is a lexical stem that can freely combine with a range of different incorporated body parts, e.g., with we ‘head’ as in (4.54b) while yekum is a lexicalized combination.

(4.56) pan-ngi-yekum

3SGslash(23).NFUT-hit.on.ear

‘He hit me on the ear (with a stick).’ (Street 1989)

Forshaw (2011) is concerned with noun incorporation in Murrinh-Patha and actually proposes a continuum between productive and lexicalized noun incorporation. He further argues that both argument and range noun incorporation is found in Murrinh-Patha. For examples such as (4.56), he argues for an external possession analysis in which the incorporated body part is a thematic object whose possessor is the direct object.

It would go beyond the scope of this work to treat incorporation in detail. However, the LCS blueprint of complex predicates with the lexical stem yekum is given as a simplified version in (4.57). In this case, the place argument is specified as AT EAR, ignoring external possession for the time being. The LCS blueprint is referred to as “$\@$HIT,EAR” and this blueprint is only used with the lexical stem yekum ‘hit on ear’. However, if more lexical stems are found that specify the place of forced contact in hitting events, a more general LCS blueprint could be proposed instead in which EAR is inserted as a further variable.

(4.57) $\@$HIT,EAR: CLASSIFIER STEM + yekum ‘hit on ear’

$$\left[ \text{CAUSE} \left( \text{[Thing]} A^\alpha, \right. \left. \text{BECOME} \left( \text{[INSTR]}, \text{[AT} \left( \text{[EAR]} \right) \text{]} \right) \right) \left[ \text{BY} \left[ \text{CAUSE} \left( \text{[Thing]} A^\alpha, \text{[AFF} \left( \text{[INSTR]}, \text{[EAR]} \right) \text{]} \right) \right] \right] \right]$$
To sum up the discussion about caused contact complex predicates, a range of different LCS blueprints have to be used in the hierarchy of selectional restrictions to account for the differences in argument structure of the complex predicates. Complex predicates of hitting and complex predicates of change of state can both be formed with the classifier stems \textit{bash}(14), \textit{poke}(19) and \textit{slash}(23), as the semantic concepts denoted by these classifier stems allow the combination with both lexical stems of hitting and lexical stems denoting a result state.

The classifier stems \textit{bash}(14), \textit{poke}(19) and \textit{slash}(23) can be used in complex predicates of movement as well, as was shown in Section 4.2.2. As has been pointed out in Chapter 3, movement can also be expressed by other classifier stems, among others the classifier stems \textit{go}(6), \textit{feet}(7) and \textit{hands}(8). Figure 4.6 presents an excerpt of the selectional hierarchy of movement complex predicates with links to LCS blueprints.

As can be seen, different LCS blueprints are used in this hierarchy as well. The use of different classifier stems mainly corresponds to the distinction between verbs of manner of motion which do not specify a path and motion verbs which denote a path. Jackendo\textsuperscript{f}ff (1990) models this distinction with the functions \textit{move} and \textit{go}, and the names of the LCS blueprints in Figure 4.6 have been chosen accordingly.

Complex predicates of manner of motion are formed with the classifier stems \textit{go}(6), \textit{feet}(7) and \textit{hands}(8). As has been discussed in Chapter 3, the classifier stem \textit{feet}(7) can be used very productively in transitive complex predicates as well as in intransitive complex predicates denoting movement. An example is repeated in (4.58a). The lexical stem \textit{barlbarl} ‘fly’ specifies the manner of motion, namely \textit{flying}. It can also combine with
the classifier stem go(6). Similarly, the classifier stem hands(8) can be used in intransitive complex predicates of one specific manner of movement, namely of gliding. An example is repeated in (4.58b).

(4.58)  a. nungammarlbirl
        nungam-barlbirl
        3SGS.FEET(7).NFUT-fly
        ‘It is flying.’ (Street 1989)

b. ma-wel-nu
    3SGS.HANDS(8).FUT-glide-FUT
    ‘It will glide.’ (Barone-Nugent 2008)

The classifier stems go(6) and feet(7) can be used in a large range of different manner of motion complex predicates. Crucially, feet(7) does not only combine with lexical stems denoting movement by feet, as can be seen in (4.58a). The LCS blueprint that accounts for both classifier stems, go(6) and feet(7), to combine with barlbarl ‘fly’, is given in (4.59). It specifies that the combinations are complex predicates of manner of motion for which the lexical stem contributes the specific manner of motion.

(4.59) @MOVE: CLASSIFIER STEM + lexical stem of manner of motion
       [ MOVE (\[Thing\]A) ]

In contrast, the complex predicates of manner of motion involving the classifier stem hands(8) use a more specific LCS blueprint, specified in (4.60). As the classifier stem hands(8) can only be used in complex predicates in which the movement is brought about by hands or hand-like body parts, the instrument by hand is added to the LCS blueprint. by hand is used here as a function specifying that an action has been brought about by means of using hands. The LCS blueprint so reflects the selectional restrictions of the classifier stem for the semi-productive combination of hands(8) with lexical stems of gliding. It also reflects the intuition that hands(8) always contributes a hand or hand-like body part as instrument to the LCS, both in complex predicates of caused contact and in complex predicates of movement.

(4.60) @MOVE,HAND: HANDS(8) + lexical stem of gliding
       [ MOVE (\[Thing\]A) BY HAND ]

The combination of the classifier stem bash(14) with the lexical stem wetuth ‘wave rolling’ can be considered a non-productive combination. According to the database, no other lexical stem of manner of movement combines with the classifier stem bash(14). Additionally, why bash(14) is used instead
of another classifier stem is not clear. Being a non-productive combination, the combinatory possibility is thus solely modeled by the hierarchy of selectional restrictions, without the restrictions being mirrored in the LCS. The combination thus just uses the LCS blueprint @MOVE.

Complex predicates of movement formed with the classifier stems poke(19) and slash(23), along with the combination of bash(14) with rikat ‘circuit’, specify a path. These combinations use the LCS blueprint @GO as specified in (4.61).

\( (4.61) \) @GO: CLASSIFIER STEM + lexical stem denoting path

\[
\text{[ GO ([Thing], [PATH]) ]}
\]

In the @GO LCS blueprint, the path is specified by the information that the lexical stem contributes. For example, the lexical stem rikat ‘circuit’ specifies a path that goes around an object. Similarly, the lexical stem rdertpart ‘skirt along edge’ contributes that the path follows an edge.

As the path is specified by the lexical stem, in these combinations it is the lexical stem that actually determines the number of arguments that the complex predicate takes. The lexical stem can either specify a path that does not need an extra argument, such as wintigat ‘descend’, or a path that normally takes an object such as riwak ‘follow’. Just as with the caused contact complex predicates, the LCS blueprints thus make visible which meaning parts are contributed by the different parts forming the complex predicates.

Additionally, the classifier stems set requirements for the path, e.g., for poke(19) that the path is required to be linear. In this way, the selectional restrictions modeled in the hierarchies of selectional restrictions are again reflected by the LCSs for productive complex predicates.

To sum up, the hierarchies of selectional restrictions determine the combinatory possibilities of classifier and lexical stem combinations, based on their compatibility of semantic concepts, not based on argument structure. Each combination of classifier and lexical stem that is compatible based on these concepts specifies which LCS blueprint it uses to realize its argument structure. While the LCS argument structure is thus the level of lexical semantics that is visible to syntax and determines which arguments are realized as grammatical functions, the hierarchies model the lexical semantics that is linked to context and world knowledge. This overall architecture is illustrated in Figure 4.7.

Despite this division of labour between selectional hierarchies and LCS blueprints, the LCS blueprints mirror the selectional restrictions of the basic concepts that restrict the Murrinh-Patha classifier and lexical stem combinations, i.e., for productive and semi-productive complex predicates. For the caused contact complex predicates, the shape of the instrument specified by the classifier stem may pose restrictions on the result state denoted by the lexical stem. Similarly, the difference between GO and MOVE models the
difference between complex predicates of manner of movement, formed with the classifier stems \texttt{GO(6)}, \texttt{FEET(7)} and \texttt{HANDS(8)} and complex predicates of movement involving a path, formed with the classifier stems \texttt{BASH(14)}, \texttt{POKE(19)} and \texttt{SLASH(23)}.

### 4.4 Conclusion

The approach proposed in this thesis for modeling Murrinh-Patha complex predicates assumes two main levels. The hierarchies of selectional restrictions model the fact that classifier stems have a rather general meaning which allows them to combine with a wide range of different lexical stems. In this system, classifier stems can have multiple different dimensions, i.e., occurring in various hierarchies at the same time, while lexical stems usually only occur in one hierarchy. The hierarchies so sort the different complex predicates into various groups and subgroups. As such, the discussion of the combinatory possibilities and the ordering into hierarchies of selectional restrictions is a relation-based approach to lexical semantics.

The majority of work on complex predicates has however been cast within a primitive-based approach to lexical semantics. Among the most important ones are approaches using LCSs. Especially Wilson’s (1999) approach to complex predicates in Wagiman has been adopted for a range of other Australian languages. However, this chapter showed that it is difficult to adapt existing LCS approaches such as Wilson’s (1999) to the Murrinh-Patha data as Murrinh-Patha classifier stems can be used in a wide variety of different complex predicates. LCSs are too fine-grained to model the use of one classifier stem in these different complex predicates with one underlying LCS. However, if multiple LCSs are defined for one classifier stem, approaches that model the compatibility purely with LCSs lose explanatory power.
Alternatively, this approach proposed to combine LCSs with hierarchies of selectional restrictions to model the argument structure and the decomposition of the meaning of Murrinh-Patha complex predicates in more detail than what is possible with the relation-based approach. The approach uses the primitive-based, decompositional approach of LCS blueprints to provide more details of the lexical semantic content of the complex predicate, i.e., to model which meaning parts are contributed by the classifier stem and which meaning parts are provided by the lexical stem. Additionally, as LCSs model argument structure, they serve as interface between the lexical semantic level and syntax. In this way, the analysis combines the strengths of both approaches, the high flexibility of the relation-based approach and the details of the primitive-based approach to lexical semantics.

That these two approaches do not have a one-to-one correspondence, i.e., that the LCS blueprints cannot be identified with one specific node of the hierarchies, seems to be a drawback of the approach at first sight. However, if a one-to-one correspondence existed, two separate levels of analysis would not be needed. As the Murrinh-Patha classifier and lexical stem can combine in many different ways, both levels and a complex correspondence between those is indeed required.

The blueprint LCSs are actually quite similar to the approach proposed by Ramchand (2008), which was presented in Chapter 2. In Ramchand’s (2008) First Phase Syntax account of Urdu complex predicates, she assumes that V can be decomposed into various phrases (initP, procP and resP) for which the light verb specifies the information of initP and procP and the result state is contributed by the main verb. In this sense, the LCS blueprint approach incorporates the same idea as Ramchand (2008) as in both cases, the main verb or the lexical stem contributes only a part of the representation of the complex predicate, e.g., in both cases only the result state. In the LCS blueprint approach, the lexical stem does not contribute a complete LCS. In the same way, in Ramchand’s (2008) approach, the main verb does not contribute a whole V decomposition that has to be merged with the V decomposition of the light verb. This idea is thus shared.

However, the two approaches differ greatly with respect to the semantic information that is assumed to be lexically relevant. While Ramchand (2008b) assumes very minimal information to be lexically relevant, LCSs as proposed by Jackendoff (1990) and consequently also the LCS blueprint approach presented here assumes that more semantic information is relevant for argument selection. These approaches thus consider information as lexical knowledge which would be considered encyclopedic knowledge by Ramchand (2008b). The Murrinh-Patha data, however, suggests that semantic concepts such as different object shapes are part of lexical information in Murrinh-Patha, as these concepts restrict the combinatory possibilities of classifier and lexical stems. The data thus speaks in favor of approaches such as Jackendoff (1990), Pustejovs’ky (1995) or Asher (2011) who assume that the lexicon contains a large amount of information.
The LCS blueprint approach can also be compared to the TCL approach proposed by Caudal et al. (2013a). Both approaches share the intuition that the meaning of a complex predicate arises from the combination of classifier and lexical stems. More detailedly, they share the assumption that it is the classifier and lexical stem together that determine what type of predicate the complex predicate is. This distinguishes them from other approach to complex predicate formation such as Wilson’s (1999) as in these accounts, the equivalent of the classifier stem determines the basic shape of the complex predicate and the equivalent to the lexical stem only provides further details for this basic shape.

Although this insight is shared by the TCL approach and the approach proposed in this work, the details of the modeling of the combinations and how the meaning arises differ considerably. In the TCL approach, classifier and lexical stems are basically treated as full verbal predicates. They have a certain event predicate type and they require a certain argument structure, just as normal verbal predicates. In complex predicate formation, the bridging functions negotiate a ‘common ground’ of the classifier and lexical stems and the additive operator combines them once this common ground is established.

In contrast, the LCS blueprint approach does not assume that classifier and lexical stems are equivalent to full verbal predicates. Most importantly, it does not assume an argument structure for the classifier and lexical stems, as this is difficult to uphold, as Chapter 3 argued. In contrast, it just assumes that classifier and lexical stems provide specific lexical-semantic specifications such as that it is an event in which a hand or hand-like body part plays a role for the classifier stem HANDS(8) or a specific result state that is specified by the lexical stem.

The LCS blueprint approach thus renders a straightforward explanation why all lexical stems and most classifier stems cannot function as sole verbal predicates. In contrast, the TCL approach does not explain why only a composition of classifier and lexical stems forms a valid verbal predicate in Murrinh-Patha (in most cases).

They also differ in how the LCS blueprints and the coercion functions work. At first glance, the LCS blueprints and the coercion functions seem to fulfill the same or highly similar functions, as both ensure that the classifier and lexical stems can combine in the required ways. The difference lies in what they assume to be fully specified in the lexicon in contrast to what might be contributed from context. LCS blueprints were defined for fixed classes of classifier and lexical stem combinations which have a lexicon entry together, specified in the hierarchy of selectional restrictions. This was useful in stressing the strict selectional properties of classifier and lexical stem combinations and their tendency for less productive or even lexicalized combinations.

In contrast, TCL as proposed by Asher (2011) stresses that bridging
functions are defeasible and can be overridden in appropriate contexts. As Section 4.1.2 argued, this might be attractive for the modeling of classifier stems of posture and motion which show a greater or lesser degree of bleaching, depending on the context. Thus, the TCL approach incorporates context sensitivity easily while it does not explain lexicalization automatically. In contrast, the LCS blueprint approach explains lexicalization straightforwardly, while it does not offer a straightforward way to incorporate context sensitivity.

Finally, an obvious difference between the LCS blueprint account and the TCL account should be pointed out. While the LCS blueprint account assumes a decompositional semantics just as other approaches to complex predicates using LCSs or First Phase Syntax, the TCL approach denies that such decompositions are necessary. However, as the Murrinh-Patha data showed, the classifier stem and the lexical stem contribute different meaning aspects to the overall meaning of the complex predicate, at least in productive combinations. This speaks in favor of a decompositional semantics, which was modeled in this work by LCS blueprints combined with hierarchies of fine-grained selectional restrictions.
An LCS blueprint analysis of complex predicates
Chapter 5

A computational implementation of the morphology

The previous two chapters described the Murrinh-Patha classifier and lexical stem patterns from the perspective of linguistic theory. The chapters could make use of the already existing resources of Murrinh-Patha, for example of the dictionary provided by Street (1989). In contrast, the following two chapters look at the Murrinh-Patha data from a computational linguistic perspective, with the aim of finding new combinations of classifier and lexical stems computationally and of developing new resources. Due to the complex morphological structure of Murrinh-Patha, this needs a careful computational implementation of the morphology.

Normally, resource development mainly focuses on well-described languages with a large amount of speakers. However, smaller languages such as Murrinh-Patha may also profit from language resources, which can then be used in corpus studies and applications such as electronic dictionaries or computer-assisted language learning materials.

Such computational resource development can, of course, highly profit from already existing theoretical resources and descriptions of the language. This is especially true for rule-based implementations such as proposed in this thesis. However, even statistical models of natural language processing may profit from detailed theoretical descriptions and a general understanding of how language works. Bender (2009) shows that a linguistically naive computational account is not automatically language independent, as is assumed in many statistical approaches. To develop truly language-independent computational models of language, Bender (2008) claims, typological knowledge is needed that allows for parameterizing the algorithms for specific language types.

Tsarfaty et al. (2013) make a similar claim for morphologically rich lan-
guages. They point out that even for this more specific language type, parameterization is still needed as morphologically rich languages themselves constitute cross-linguistic variation that needs to be modeled in diverse ways in computational implementations. Tsarfaty et al. (2013) name a range of dimensions along which morphologically rich languages may differ and which may be important for computational analysis. To these dimensions belong the richness and complexity of the morphology, the flexibility of word order as well as the degree of syncretism that is exhibited by morphological markers. A computational implementation for one morphologically rich language may thus face very different problems from an implementation of another morphologically rich language. Tsarfaty et al. (2013) are mainly concerned with statistical models of language processing. However, this claim is naturally also true for rule-based approaches. Finally, the development of resources for smaller languages like Murrinh-Patha may face the challenge that not enough data is available for a successful statistical approach or that the language is not described detailedly enough.

Murrinh-Patha offers itself as a case study for a computational implementation due to its complex verbal structure, its free word order and the theoretical descriptions of the language that are available. This chapter focusses on the implementation of the Murrinh-Patha morphology, its challenges for the implementation and on the question of how a morphological analyzer can be used to gain new insights into the data.

The complex Murrinh-Patha morphology is challenging for a computational implementation, as is discussed in Section 5.1. However, as Section 5.2 shows, a computational implementation of the morphology can be achieved with the help of the Xerox finite-state technology tools XFST and LEXC (Beesley & Karttunen 2003). These tools offer mechanisms such as flag diacritics that can be used to model the details of the complexities as closely as required.

The implementation is then tested on a small corpus of Bible translations (approximately 70800 words) to see how well the implementation is able to analyze real text. Although Murrinh-Patha has received considerable attention from linguists in recent years, e.g. Street (1987, 1989), Walsh (1987, 1995, 1996, 1997), Blythe (2009a), Nordlinger (2008, 2010c, 2011, 2015), it cannot be presumed that all phenomena have been described in detail yet. A corpus study can help identify unknown phenomena and lexical items.

The Murrinh-Patha corpus study described in this chapter concentrates on unknown morpheme orderings and new lexical items. The identification of these is achieved with a morphological analyzer with a stepwise lookup strategy. The stepwise analyzer gives priority to analyses for already known morpheme orderings and known lexical items, but can analyze unknown morpheme orderings and new lexical items as well. As such, it can be used to extract candidates for new lexical items that can be tested in future fieldwork. Similarly, the undocumented morpheme orderings should also be
tested in future fieldwork to see whether these are indeed grammatical or whether these are mistakes in the corpus.

Sections 5.3 and 5.4 are concerned with this corpus study. Section 5.3 presents the development of the stepwise morphological analyzer in which different lookup strategies are applied after each other. In addition to the first strategy covering well-documented phenomena, the stepwise morphological analyzer comprises a strategy that covers undescribed morpheme orderings, a strategy for unknown classifier and lexical stem combinations and a morphological guesser for lexical stems.

The stepwise morphological analyzer thus allows to identify candidates for new Murrinh-Patha verbs. These are analyzed in detail in Section 5.4. Section 5.4.1 presents the results of the lookup strategy with relaxed constraints on classifier and lexical stem combinations while Section 5.4.2 deals with the results of the fourth lookup strategy, i.e., of the morphological guesser for the lexical stems. The sections also discuss how the new verb candidates can be interpreted based solely on the corpus study. Section 5.5 summarizes the chapter.

5.1 Challenges posed by the Murrinh-Patha verbal template

As could be seen throughout the thesis so far, the Murrinh-Patha verbal template is highly complex. This poses special challenges to a computational treatment of the morphology. These challenges can be sorted into two different groups: firstly, a range of properties of the Murrinh-Patha verbal morphology results in a high number of very different surface forms in a given text. This means that a morphological analysis is needed to abstract away from the actual surface form to extract the relevant factors. Secondly, the templatic nature of the verbal template, which was discussed by Nordlinger (2010c), poses special challenges to the implementation of a morphological analyzer. This section looks at the complexities of the Murrinh-Patha verb as presented in Chapter 1 in more detail and especially interprets these complexities in relation to the challenges they pose for a computational treatment of the morphology.

As was discussed in Chapter 1, the Murrinh-Patha verb may consist of up to nine different morphemes. The verbal template is repeated in Table 5.1. As has been discussed in Chapter 3 and 4, the lexical meaning of the verb is mostly determined jointly by the classifier stem in slot 1 and the lexical stem in slot 5. Only a small range of classifier stems can function as the sole verbal predicate, which means that most Murrinh-Patha verbs consist of at least two different morpheme, the classifier stem and the lexical stem.

1Earlier versions of this section were published as Seiss (2011, 2012b).
As has been also pointed out in Chapter 1, classifier stems inflect for tense as well as for subject number and person. The inflection on the classifier stem is encoded in portmanteau forms, an example of the verbal paradigm of the classifier stem see(13) was provided in Table 1.2 and is repeated in Table 5.2. The inflection results in more than 50 different forms per paradigm, i.e. in forms which have a different surface realization although underlyingly, the same classifier stem is used. (5.1) provides examples with the classifier stem see(13) in combination with the lexical stem ngkardu ‘see’.

(5.1) a. bam-ngkardu
    1SGS.SEE(13).NFUT-see
    ‘I saw him/her.’

b. ngube-ngkardu-dha
    1PLS.SEE(13).PIMPF-see-PIMPF
    ‘We saw him/her.’

Other verbal template slots contribute to variation in the surface form of the verb, too. Body parts may be incorporated in slot 4 and adverbials and particles may be attached in slot 7 and 9.

In addition, phonological processes may apply when morphemes are combined which disguise the fact that different surface realizations may involve the same underlying morpheme. For example, in (5.2), the same lexical stem ngkardu ‘see’ is used. However, as was described by Street (1987), the nasal /ng/ is lost in (5.2a) because of the preceding /m/ of the classifier stem. In
contrast, the nasal is retained in (5.2b) which involves the same classifier stem but which has an intervening subject number marker between classifier and lexical stem. As in this case the nasal /ng/ is not preceded by /m/, it is retained.

(5.2) a. *bam*<sub>kardu</sub>
        bam-ngkardu
        3SGS.SEE(13).NFUT-see
        ‘He/she saw him/her.’

       b. *bamnginthang*kardu
        bam-ngintha-ngkardu
        3SGS.SEE(13).NFUT-DU.F-see
        ‘They two (non-siblings) saw him/her.’

The nasal /ng/ is always lost after /m/ or /n/ (Street 1987). However, Street (1987) also discusses more complex rules in which the application of the rule depends on the lexical stem used. This is illustrated in (5.3) in which the sequence of /m/ and /y/ is changed to /nth/ for the lexical stem *yerr* ‘look’ but not for the lexical stem *yet* ‘rain’.

(5.3) a. *mintherr*
       mim-yerr
       1SGS.12.NFUT-look
       ‘I looked out/around.’  

       (Street 1987)
b. *kanamyel*
   kanam-yel
   3SGS.BE(4).NFUT-rain
   ‘It is raining.’ (Street 1987)

Phonological processes such as these are very common in Murrinh-Patha. The phonological processes, the number of morphemes in the verbal template as well as the many different forms of the classifier stems make a surface-orientated analysis of Murrinh-Patha text difficult. Therefore, a morphological analyzer is needed that abstracts away from the surface form. Such a morphological analyzer needs to be able to implement the phonological processes found in Murrinh-Patha. Additionally, the analyzer has to face another challenge that is posed by the Murrinh-Patha verb: long distance dependencies between the different morphemes in the verbal template. In the following, a range of different long distance dependencies found in the Murrinh-Patha verbal template are discussed.

As was discussed extensively in Chapter 3 and 4, the lexical meaning of the verb is determined jointly by two morphemes, the classifier stem in slot 1 and the lexical stem in slot 5. As not every classifier stem can combine with every lexical stem and vice versa, a long distance dependency between these slots exists.

It is important to model the dependency between these template slots because of the high degree of syncretism found in the paradigms of the classifier stems. As can be seen in (5.4), the surface form bam can be classifier stem 13 or 18. However, only the classifier 18 can combine with the lexical stem bat ‘fall’. Thus, in this case, bam can only be interpreted as classifier 18.

(5.4) *bam-bat*
   1SGS.18.NFUT-fall / *1SGS.SEE(13).NFUT-fall
   ‘I fell.’

A morphological analyzer should thus model the dependency between classifier and lexical stems as otherwise, it would return both analyses and therefore overgenerate.

A similar long distance dependency can be found for tense marking, affecting the template slots 1 and 6. As has been stated above, the classifier stems inflect for tense. Additionally, as was discussed by Nordlinger & Caudal (2012), all tenses, except the non-future and future irrealis tense, require an additional tense marker in slot 6 which agrees with the tense marking on the classifier stem. For example, the verb in (5.5a) needs a future marker in slot 6 because the classifier stem is in future tense. No tense marker, or an alternative tense marker, e.g., a past imperfective marker as in (5.5b), is ungrammatical.
While the dependencies involving classifier and lexical stems and tense marking can be considered as simple agreement and can be modeled as simple long distance dependencies in the implementation, more complex dependencies exist as well. Nordlinger (2010c) discusses these interdependencies in detail. Here, only some examples involving subject number and object markers are discussed.

The relationship between the number information encoded in the classifier stem and the separate subject number markers in slots 2 and 8 represents a complex long distance dependency. As has already been discussed, the classifier stems themselves show a three-way number contrast: singular, dual and plural. The singular classifier stem can combine with a separate dual number marker which overwrites the information of the classifier stem, as can be seen in (5.6a). The dual classifier stem without a separate subject number marker denotes the dual, sibling category as in (5.6b). However, this information can also be overwritten by attaching a paucal number marker as in (5.6c).

As can already be seen in these examples, the number markers for dual and paucal subject behave differently in the verbal template. The paucal number marker attaches after the lexical stem (in slot 8) while the dual number marker attaches before the lexical stem (in slot 2). However, the placement of the dual number marker is more complex as can be seen in the examples in (5.7). The dual subject number marker competes with the object markers for slot 2. In (5.7a), the dual female subject number marker -ngintha attaches in slot 2, between the classifier and lexical stem. However, if an object marker is present as in (5.7b), the subject number marker has to be realized in slot 8, after the lexical stem. As (5.7c) shows, this is impossible if the object marker is not expressed in slot 2.

(5.5)    a. ba-ngkardu-nu
         1SGS.SEE(13).FUT-see-FUT
         ‘I will see him/her.’

   b. *ba-ngkardu-dha
         1SGS.SEE(13).FUT-see-PIMPF

(5.6)    a. bam-ngintha-ngkardu
         3SGS.SEE(13).NFUT-DU.F-see
         ‘They two (non-siblings) saw him/her.’ (Nordlinger 2010c)

   b. pubamka - ngkardu
         3DU.S.SEE(13).NFUT-see
         ‘They two (siblings) saw him/her.’ (Nordlinger 2010c)

   c. pubamka - ngkardu - ngeme
         3DU.S.SEE(13).NFUT-see-PAUC.F
         ‘They few (non-siblings) saw him/her.’ (Nordlinger 2010c)
These examples make clear that it not only depends on whether a morpheme is present or not, it also depends on the relative position of the morphemes to each other. This has lead to the claim that the Murrinh-Patha verbal morphology is a templatic system (Nordlinger 2010c).

Another example of such a complex interplay is presented in (5.8), which illustrates that the dependencies can also help to disambiguate expressions in certain cases. The dual morphemes -*ngintha/-*nintha can be both used for subject and object number marking. As has already been pointed out, -*ngintha/-*nintha can attach to singular classifier stems to denote a dual non-sibling subject. This is illustrated in (5.8a). The morphemes -*ngintha/-*nintha can, however, be also used to denote object number marking, for example in combination with the morpheme -*nganku, which denotes a dual sibling direct object when used on its own. In combination with the morphemes -*ngintha/-*nintha, however, -*nganku denotes a dual female/male non-sibling direct object. The two interpretations of (5.8b) thus evolve from the interpretation of -*ngintha/-*nintha as a subject or object marker. In the first reading, -*ngintha is interpreted as an object marker, forming a dual female non-sibling object with -*nganku. In the second interpretation, -*ngintha is interpreted as a subject marker, denoting a dual female non-sibling subject together with the singular classifier stem. In contrast, -*ngintha cannot denote a subject number marker in (5.8c), as the classifier stem is not singular in this case.\(^2\)

\(^2\)The flexibility in interpretation of the subject in (5.8c) stems from the fact that pubam can be both a dual and a plural classifier stem if slot 2 is occupied. For details of this dependency, see Nordlinger (2010c).
c. pubam-nganku-ngkardu-ngintha
   3DUP.3.SEE(13).NFUT-1DU.SIBL.DO-SEE-DU.F
   ‘They (dual sibling, paucal sibling or plural) saw us (dual female non-sibling).’

The examples in (5.7) and (5.8) show that a detailed modeling of the interdependencies is helpful. In (5.7), the detailed modeling excludes impossible combinations and may so restrict overgeneration. In the case of (5.8), the detailed modeling excludes unwanted interpretations, which may be helpful when the examples are looked at from a syntactic or semantic point of view. A morphological analyzer should thus be able to model these kinds of complex interdependencies.

In contrast to the Murrinh-Patha verb, nominal expressions such as nouns, noun classifiers and pronouns do not pose special challenges to the implementation. They may be case-marked and inflected with discourse markers, as can be seen in (5.9). The noun classifier *kardu* is marked with the focus marker *-ka*, the pronoun *nukunu* with the dative case marker *-nu* and the noun *kunginire* ‘yesterday’ with the discourse marker *yu*. This nominal marking does not pose special challenges to the computational implementation of the morphology as such affixation is found in many languages and computational morphologies are prepared to model such inflection.

(5.9)  

\[
\begin{array}{l}
\text{kardu-ka pardi-na-rel-dha} \\
\text{NCHUMAN-FOC 3PLS.BE(4).PIMPF-3SG.BEN-sing-PIMPF} \\
\text{nukunu-nu kunginire-yu} \\
\text{3SG.M-DAT yesterday-DM}
\end{array}
\]

‘They were singing for him yesterday’ (Nordlinger 2011)

For this reason, the computational implementation of the morphology as presented in the following section concentrates on the challenges the Murrinh-Patha verb poses. A computational implementation has to offer tools to model the phonological processes as well as the complex long distance dependencies. A careful modeling of these complexities is needed to restrict overgenerating of the morphological analyzer.

### 5.2 Implementation of a finite state morphological analyzer

In the previous section, the difficulties that Murrinh-Patha poses for a computational morphological treatment were summarized. The complexities of the Murrinh-Patha verbal inflection require a morphological analysis to

---

3 Earlier versions of this section were published as Seiss (2011, 2012b).
abstract away from the surface realization. In order to achieve such a morphological analysis, a morphological analyzer needs to be implemented which can handle the long distance dependencies and the phonological rules that apply. Such a morphological analyzer was implemented using the Xerox finite-state technology tools Xfst and Lexc (Beesley & Karttunen 2003). This system has been used because it offers a wide range of inbuilt mechanisms to handle complex cases such as those found in the Murrinh-Patha verb. This section discusses how these inbuilt mechanisms facilitate the modeling of the complexities on some selected examples. The following section then describes how this implementation is used to construct a robust morphological analyzer that can be successfully applied to real text.

Before the details of the implementation of the morphological analyzer are discussed, though, a brief overview over related work, i.e., over rule-based morphological analyzers using finite state methods for more or less morphologically complex languages, is provided.

The concept of finite state morphology was developed in the 1980s as a tool for the computational morphological analysis of natural language. Beesley & Karttunen (2005) provide a detailed historic overview and a formal description of the formalisms. Since Koskenniemi (1983) combined ideas of sequenced phonological rewrite rules with two-level morphology and proposed a first computational implementation of Finnish morphology, finite state morphologies have been developed for a number of diverse languages, language families and language types. Some examples are the treatment of South Asian languages such as Bögel et al. (2007) for Urdu or Veerappan et al. (2011) for Kannada, the treatment of Indonesian by Larasati et al. (2011) as a representative for Austronesian languages or for Estonian (Uibo 2005) and Finnish (e.g., Lindén & Pirinen 2009) from the Uralic language family.

Semitic languages have traditionally been in the focus of finite-state methods for the treatment of their morphology due to the special challenges the root-and-pattern morphology poses. Some examples of treatments of the Semitic languages are, e.g., Beesley (1996) and Attia et al. (2011) for Arabic as well as Yona & Wintner (2008) for Hebrew.

Karttunen (2003) presents an Xfst implementation of the Bantu language Lingala in which he shows that even analyses cast within paradigm function morphology (Stump 2001) can be implemented with Xfst’s regular expressions.

While all these approaches face different challenges posed by the respective morphology of the language, to my knowledge so far no finite state implementation of an Australian language with such a complex templatic verbal morphology as is found in Murrinh-Patha has been proposed. Sproat & Brunson (1987) present a morphological analyzer of Warlpiri, focusing on the problem of reduplication. They make heavy use of prosodic information. Their implementation does not use a two-level morphology and is also not
finite-state. As such, the basic layout of the analyzer differs considerably from the Murrinh-Patha analyzer presented here and the other finite-state analyzers mentioned above.

Closest to the implementation of Murrinh-Patha described here is probably the treatment of Basque by Alegria et al. (1996) and the treatment of Persian by Megerdoomian (2004). The implementation of the Persian morphology proposed by Megerdoomian (2004) is concerned with problems involving tokenization, phonological rules and long-distance dependencies. However, it seems that the long-distance dependencies for Persian, which are modeled with flag diacritics (see below), are simpler than the ones found in Murrinh-Patha.

Alegria et al. (1996) in their implementation for Basque propose a formalism for long-distance dependencies, but treat only simple long-distance dependencies. Their implementation is, however, relevant because it proposes a three-step lookup strategy similarly to the lookup strategy proposed in the following section.

Most recent projects either use xfst (Beesley & Karttunen 2003), or the open-source alternatives Foma (Hulden 2009) or HFST (Helsinki Finite-State Transducer, Lindén et al. 2011). XFST is, for example, used in the implementations by Yona & Wintner (2008) for Hebrew or by Bögel et al. (2007) for Urdu. On the other hand, Attia et al. (2011) and Larasati et al. (2011), for example, explicitly state that they use Foma because of licensing issues.

For the implementation of Murrinh-Patha that is described in this chapter, the Xerox finite-state technology tools XFST and LEIX (Beesley & Karttunen 2003) have been used, as the combination of both programming languages provides powerful tools to implement morphological complexities. The goal of the computational implementation is to associate a surface form of a word with its morphological analysis. An example is provided in (5.10) in which the Murrinh-Patha verb *bamkardu* is associated with the analysis as the classifier stem *see(13)* (+class13) in 3rd person singular non-future form, a 3rd person singular direct object and a lexical stem *ngkardu*.

(5.10) bam+class13+3P+sg+nFut+3sgDO+ngkardu+LS : bamkardu

For such a pair, the expressions upper and lower level are used. The morphological analysis (*bam+class13+3P+sg+nFut+3sgDO+ngkardu+LS*) is the upper level while the surface form *bamkardu* is the lower level. The computational implementation is bidirectional, i.e., if presented with the analysis as input, it can return the surface form as output and vice versa.

To model these levels, different mechanisms offered by XFST and LEIX are used. The concatenation of morphemes is implemented with LEIX (Beesley & Karttunen 2003) as two-level networks. It uses continuation classes which are implemented as so-called lexicons. A first simple example is provided in
The first lexicon is called ROOT, it comprises all possible first morphemes of a word. In this lexicon, the classifier stem \( \text{bam} \) is associated with the morphological information that it carries \((\text{bam+class13+3P+sg+nFut})\). The right side of the entry specifies which lexicon is used next. In (5.11), \( \text{bam} \) can be concatenated with objects from the lexicon SLOT2, which in this case only contains the 3rd person direct object marking which is not overtly realized (noted as 0). This combination combines with items from the lexicon LEX, which contains the lexical stem \( \text{ngkardu} \). The hash key marks the end of a word.

(5.11) Lexicon ROOT
\[ \text{bam+class13+3P+sg+nFut}:\text{bam} \quad \text{SLOT2}; \]
Lexicon SLOT2
\[ +3\text{sgDO}:0 \quad \text{LEX}; \]
Lexicon LEX
\[ +\text{ngkardu+LS}:\text{ngkardu} \quad #; \]

In this way, the network in (5.11) can produce the output in (5.12). The string \( \text{bamngkardu} \) is associated with the information on the upper side, i.e., that this word is made up of a classifier stem \( \text{bam} \) which is classifier stem 13 inflected for third person singular non-future tense, a zero direct object marker and a lexical stem \( \text{ngkardu} \).

(5.12) \( \text{bam+class13+3P+sg+nFut+3sgDO+ngkardu+LS} : \text{bamngkardu} \)

In the actual implementation of Murrinh-Patha verbs, the lexicon ROOT contains all forms of the 38 classifier stems, the lexicon SLOT2 all different object morphemes (among others), and a large number of different lexical stems are contained in the lexicon LEX. The other template slots are implemented with the help of lexicons in a similar way.

However, as was discussed above, the surface forms of Murrinh-Patha verbs are often more than pure concatenations of morphemes due to the phonological rules that apply when morphemes are combined. XFST offers the possibility to formulate phonological rules which rewrite the concatenation of morphemes to the actual surface form of the verb (Kaplan & Kay 1994). For example, (5.13) accounts for the data in (5.2) in which /ng/ is lost if it follows an /m/ or /n/. Thus, the actual network contains the surface form and associated information as specified in (5.10).

(5.13) \[ n \ g \ k \rightarrow k \ || \ m , \ n \ ]

For more complex cases in which the application of the rule depends on the lexical stem as in (5.3) above, the lexical stem can be marked when it is concatenated with the other morphemes and the regular expression
can take this marking into account. Thus, the output of the concatenation of morphemes for the data in (5.3) is as illustrated in (5.14). In (5.14a), the lexical stem *yel* does not trigger any phonological change. In contrast, the lexical stem *yerr* in (5.14b) triggers a change and is therefore marked especially with a capital letter /Y/. The application of the phonological rules in (5.15) then causes /Y/ to be changed to /nth/ after /m/ and to /y/ in all other cases and so ensures the right surface form. The final network for these two examples associates the morphological information with the surface forms *kanamyel* and *mintherr* as is displayed in (5.16).

(5.14)  
\[ a. \text{kanam+class4+nFut+3P+sg+nFut+yel+LS} : \text{kanamyel} \]
\[ b. \text{mim+class12+nFut+1P+sg+nFut+yerr+LS} : \text{mimYerr} \]

(5.15)  
\[ \text{[ m Y – > nth ] .o. [ Y – > y]} \]

(5.16)  
\[ a. \text{kanam+class4+nFut+3P+sg+nFut+yel+LS} : \text{kanamyel} \]
\[ b. \text{mim+class12+nFut+1P+sg+nFut+yerr+LS} : \text{mintherr} \]

The two-level approach makes it possible to distinguish between surface form and the associated information. This is especially helpful for Murrinh-Patha as the same classifier and lexical stem combination might have very different surface forms, due to the large number of different forms of the classifier stems, the high number of morphemes in the verbal template and the phonological rules that apply. The two-level morphology allows one to abstract away from this surface form. This, for example, makes a statistical analysis of texts independently of the surface form possible.

Murrinh-Patha is, however, challenging to a computational morphological implementation due to its long distance dependencies found in the verbal template. Dependencies between neighboring lexicons can be easily modeled by specifying different continuation classes, i.e., entries of one lexicon do not have to lead to the same next lexicon. However, most dependencies in the Murrinh-Patha verbal template are long-distance, which is very difficult to model just with continuation classes. *Xfst* offers a possibility to model these long-distance dependencies with the help of flag diacritics.

Flag diacritics are special entities in *Xfst* which add a kind of “short term memory” to keep track of what choices have been made before. Thus, as Beesley & Karttunen (2003:341) explain, normally, “the transition from one state to the next depends only on the current state and the next input symbol”. Using flag diacritics allows one to keep track of choices made earlier, so that certain transitions can also be constrained by choices made earlier.

In the implementation, flag diacritics can be recognized by two surrounding @-symbols. After the first @-symbol, an operator is followed by a feature-value pair, each separated by periods. Different operators exist,
i.e. U(nification), P(ositive) setting, R(equire) test, D(isallow) test etc. The names of the features and values can be chosen arbitrarily, but for convenience, mostly morphological features and values have been chosen.

As a first simple illustration of the use of flag diacritics, the long distance dependency between the tense marking on the classifier stem and separate tense markers in slot 6 will be discussed. For all tenses but the non-future and future irrealis tense, tense markers in slot 6 are obligatory. The relevant examples are repeated in (5.17). In example (5.17), \textit{bam} is the non-future form of the classifier stem 13 while \textit{ba} is the future form of the corresponding classifier. The future form has to combine with the future tense marker \(-nu\) (tagged as \(+\text{Fut2}\)) as can be seen in (5.17b); it is ungrammatical without \(-nu\) ((5.17c)). On the other hand, \(-nu\) cannot attach to the non-future classifier stem form ((5.17d)).

\begin{equation}
\text{(5.17)}
\begin{align*}
a. \quad & \text{bam} + \text{class13} + 3P + sg + nFut + 3sgDO + ngkardu + LS: \\
& \text{bam-ngkardu} \\
b. \quad & \text{ba} + \text{class13} + 3P + sg + Fut + ngkardu + LS + Fut2: \\
& \text{ba-ngkardu-nu} \\
c. \quad & \text{ba} + \text{class13} + 3P + sg + Fut + ngkardu + LS: \ast \text{ba-ngkardu} \\
d. \quad & \text{bam} + \text{class13} + 3P + sg + nFut + 3sgDO + ngkardu + LS + Fut2: \\
& \ast \text{bam-ngkardu-nu}
\end{align*}
\end{equation}

This interplay can be modeled with the help of P- and R-type flag diacritics. The P-type flag diacritics are used to set a value, e.g., for tense, to positive. In contrast, the R-type flag diacritics require a value to have been set to positive to allow the respective combination. (5.18) is a fragment of the implementation. The lexicon \textsc{ROOT} lists the classifier stems. From this lexicon, the various classifier stem forms are sent to different lexicons to pick up their respective tense information. For example, \textit{bam} carries non-future information and is consequently sent to the lexicon \textsc{NFUT} to pick up the flag diacritic \(\ast @P.Tense.nFut\)’. This flag diacritic sets the value for the attribute ‘Tense’ to ‘\text{nfut}’, i.e., it remembers that \textit{bam} is non-future tense. Similarly to \textit{bam}, other non-future classifier stem forms are also sent to the same lexicon \textsc{NFUT}. In contrast, classifier stem forms in future tense such as \textit{ba} are sent to the lexicon \textsc{FUT} and receive the flag diacritic \(\ast @P.Tense.Fut\)’ to remember that its tense value is future.

After the flag diacritics for tense information are picked up, other morphemes from the verbal template slots 2 to 5, e.g., direct and indirect object markers, incorporated body parts, lexical stems etc., can be attached. This is not represented in detail in (5.18) but indicated by the dots. Flag diacritics are used instead of continuation classes in this case because many morphemes can intervene between the classifier stem inflected for tense and the tense markers in slot 6.

Finally, when the corresponding tense markers are attached in the lexicon \textsc{TENSE\_SLOT6}, the choices for the combination are constrained by the
R-type flag diacritics which require a certain value for tense. The morpheme -nu can only attach to a future classifier stem form, i.e., this choice is marked with the flag diacritic "@R.Tense.Fut@" and consequently, only strings are possible which include the flag diacritic "@P.Tense.Fut@". Similarly, the first line in the lexicon TENSE_SLOT6 is marked by the flag diacritic "@R.Tense.nFut@" which specifies that this choice, i.e., no morpheme attaching in this slot, is only possible if the value of the feature “Tense” has been set to “nFut” before.

(5.18) Lexicon ROOT
bam     NFUT;
ba      FUT;

Lexicon NFUT
@P.Tense.nFut@ LEX;

Lexicon FUT
@P.Tense.Fut@ LEX;
...
...

Lexicon TENSE_SLOT6
@R.Tense.nFut@ #;
@R.Tense.Fut@:nu@R.Tense.Fut@ #;

The dependencies between the classifier and lexical stems can be modeled in a similar way. They again need flag diacritics because the dependencies are long distance, between the verbal template slot 1 and 5. (5.19) shows an excerpt from the LEXC lexicons. The classifier stems stand(3) and hands(8) can (among others) combine with the lexical stem dharday ‘down’. To model the long distance dependencies, flag diacritics are used to remember which classifier stem has been chosen. A separate flag diacritic is used for each classifier stem. This also means that the lexical stem has to be listed multiple times, i.e., for each combination with a classifier stem.4

4Seiss (2011) argues for modeling the combinations of classifier and lexical stems in the sublexical entries of xle. While this is desirable from a theoretical perspective as the restrictions on the combinations are then bound to argument structure information, from a computational implementation perspective, modeling the dependencies in the xfst morphology already has a range of advantages: The morphology can then be used as a stand alone application, for example in a corpus study, as is described in Section 5.3.
These dependencies, i.e., the dependencies for tense markers and the dependencies between classifier and lexical stems, are quite simple examples of long distance dependencies. However, flag diacritics also allow the modeling of complex long distance dependencies such as the subject number and object marker dependencies which are dependencies between three different verbal template slots, as they affect slot 1 for the classifier stem as well as slots 2 and 8 for the object and number markers.

As was discussed above, the subject number markers and the object markers compete for the same slots, i.e., for slots 2 and 8. For example, the dual subject number marker has to attach in slot 2 if no object marker is present. If an object marker is present, the object marker has to attach in slot 2 and the subject number marker can only be realized in slot 8. This was illustrated with the examples in (5.7). These facts concerning the singular classifier stem can be modeled with the help of flag diacritics as displayed in the implementation fragment in (5.20).

(5.20) Lexicon ROOT

bam +class13...+sg @P.NUM.sg@...:bam @P.NUM.sg@

Lexicon SLOT2

@P.SMark.no@
+1sgDO:ngi
+du.m.Nsibl.S @P.SMark.pres@@R.Num.sg@:
intha @P.SMark.pres@@R.Num.sg@

Lexicon SLOT8

+du.m.Nsibl.S @D.SMark.pres@@D.SMark.no@@R.Num.sg@:
intha @D.SMark.pres@@D.SMark.no@@R.Num.sg@ #;

In the lexicon ROOT, the classifier stem form *bam* is associated with the singular form of classifier 13, and this choice is marked by the P-type flag diacritic, i.e., it remembers that the value for the number feature singular has been set positively.

In the lexicon SLOT2, three different choices are possible. In the first case, nothing is attached. This is for example the case for intransitive verbs with singular subjects. However, the system has to remember that nothing has been attached in this slot, which is implemented with the flag diacritic.
Alternatively, an overtly expressed object marker can attach in slot 2, e.g. the marker for the first person singular direct object marker -ngi. As a third choice, the dual male non-sibling subject number marker -nintha can attach in slot 2. However, -nintha can only attach if the classifier stem form is singular, which is modeled by the flag diacritic @R.Num.sg@, which requires the value of the number feature to have been positive before. In this case, the flag diacritic @P.SMark.pres@ tells the system to remember that the dual subject marker is present in slot 2.

The lexicon SLOT8 then takes care of all possible choices. Thus, the dual subject number marker can only attach in slot 8 if it is not present in slot 2. This dependency is modeled by the flag diacritic @D.SMark.pres@ which disallows this choice if the value of the feature SMark has been set to “pres(ent)” before. Secondly, the dual number marker can only attach in slot 8 if slot 2 is not empty, i.e. this choice is disallowed if the value of the SMark has been set to “no” before. And thirdly, as has been already discussed before, the classifier stem has to be in singular form.

The actual implementation is, however, even more complicated as subject and object number morphemes also compete for slot number 8, as has been discussed above for the example (5.8). That is, nintha/ngintha can also mark object number if a dual marker such as nganku is present in slot 2. This can be modeled accordingly with flag diacritics as well, as can be seen in the more detailed implementation fragment in (5.21).

(5.21) Lexicon ROOT

\[
\text{bam+class13...+sg@P.NUM.sg@..:bam@P.NUM.sg@} \quad \text{SLOT2;}
\]

Lexicon SLOT2

@P.SMark.no@ \quad RR;

+1sgDO:ngi \quad RR;

+du.m.Nsib1.S@P.SMark.pres@@R.Num.sg@

: nintha@P.SMark.pres@@R.Num.sg@ \quad RR;

+1duDO@P.DOMark.du@:nganku@P.DOMark.du@ \quad RR;

...

Lexicon SLOT8

+du.m.Nsib1.S@D.SMark.pres@@D.SMark.no@@R.Num.sg@

: nintha@D.SMark.pres@@D.SMark.no@@R.Num.sg@ #;

+du.m.Nsib.O@R.DOMark.du@

: nintha@R.DOMark.du@ #;

+du.m.Nsib.O@R.DOMark.du@

: nintha@R.DOMark.du@ ZEROSMARK;

Lexicon ZEROSMARK

+du.m.Nsib1.S@R.Num.sg@

:@R.Num.sg@ #;

If a dual object marker such as nganku is attached in SLOT2, it is marked
with the flag diacritic @P.DOMark.du@. In the lexicon SLOT 8, the interpretation of *nintha* as an object marker is only allowed if a dual object marker has been attached in SLOT 2. This is implemented with the flag @R.DOMark.du@. In this case, then, the dual subject marker does not have a place to attach overtly in the verbal template and the example is ambiguous. This is modeled with the addition of another lexicon ZEROSMARK, in which the tag +du.m.Nsibl.S is attached if the classifier stem is in singular number (@R.Num.sg@).

5 The advantage of modeling these interdependencies in as much detail as possible is that the resulting network just represents the valid combinations of verbal morphemes and does not overgenerate. This is especially important because of the large degree of syncretism which would, without a detailed modeling of the dependencies, lead to many different analyses for one item. The inbuilt mechanisms provided by LEXC and XFST thus allow a reliable model of the details of the complexities of the Murrinh-Patha verb. The following section describes how these mechanisms can be put to use in a robust morphological analyzer with a stepwise lookup strategy.

5.3 Developing a robust morphological analyzer with a stepwise lookup strategy

The previous section described the details that are needed if one implements such morphologically complex languages and the tools that XFST and LEXC offer for these details. This section now discusses how these implementations can be put to use in a robust morphological analyzer in a corpus study of Murrinh-Patha. For this aim, a stepwise lookup strategy is used that relaxes the constraints of the verbal template stepwise and also includes a morphological guesser. In this way, both, high coverage and precision, can be achieved.

The implementation of a morphological analyzer usually starts out with the collection of the relevant facts, i.e., in the case of the Murrinh-Patha morphological analyzer, the description of the verbal template and the inflection for the other parts of speech were studied. For this purpose, it mainly used Street (1989) and Joe Blythe’s toolbox dictionary6 as a database for lexical items as well as Nordlinger (2010c) for the description of the verbal template and Blythe (2009a) for nominal inflection. As such, the computational implementation profits from the rather substantial theoretical description existing for Murrinh-Patha.

However, these resources are limited nonetheless. Especially for the bigger syntactic categories such as nouns and lexical stems it has to be assumed

---

5 The object marker *nganku* can also combine with *ngime/nime* to mark paucal nonsibling objects. This is modeled in the same manner as the dual interpretation.

6 http://www.sil.org/computing/toolbox/
that the database cannot cover all existing lexical items. For example, approximately 1040 different lexical stems and 1250 different nouns were listed in the database. It can be expected that more lexical items exist for these categories.

For this reason it makes sense to test the morphological analyzer on a larger collection of real text instead of selected test sentences or words. In this way, non-expected lexical items and constructions may be spotted which can then be analyzed, tested in fieldwork and included in the implementation. For the development of the Murrinh-Patha morphological analyzer, a small Bible corpus has been used. The translations of the Bible chapters comprise around 70800 words.

In order to make predictions of how well a computational analysis may work on unexpected texts, it is common in computational linguistics to divide a corpus into a development corpus and a test corpus. The output of the development corpus is analyzed during the development of the morphological analyzer and new insights that the output generates are implemented in the analyzer. Once the development of the morphological analyzer based on the development corpus is finished, the morphological analyzer is run over the test corpus. The results of the test corpus then indicate how well the morphological analyzer does on unknown text.

For the corpus study of Murrinh-Patha, the Bible corpus has been divided into a development corpus (approximately 85%) and a test corpus (approximately 15%). Thus, the development corpus consists of around 60080 words and the test corpus of around 10720 words.

As these corpora comprise running text, many word forms of course occur multiply in the corpus. This is especially the case for nominals and other lexical items which can be inflected for case and discourse marking, but which do not show the same high morphological complexity as verbs. The most extreme example is the conjunction i ‘and’ which occurs 2416 times in the whole corpus. Running the morphological analyzer over the text as it is provides a morphological analysis for each word as often as it occurs in the corpus.

However, the morphological analyzer can also be tested on word lists that are extracted from the running text. In these word lists, each different surface form only occurs once. That is, the conjunction i ‘and’ is only once in the word list and thus receives a morphological analysis only once. In the corpus study presented in this chapter, word lists were extracted both for the development and the test corpus. The development corpus rendered a word list of 5626 types and the test corpus a wordlist of 1643 types. Table 5.3 provides an overview over the various versions of the Bible corpus used in the development and evaluation of the morphological analyzer.

The results of the word lists and the running text complement each other. The results of the word lists show which surface forms can be given an analysis. It however does not make a distinction between high and low
A computational implementation of the morphology

<table>
<thead>
<tr>
<th></th>
<th>Running text</th>
<th>Word lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development corpus</td>
<td>60080 words</td>
<td>5626 types</td>
</tr>
<tr>
<td>Test corpus</td>
<td>10720 words</td>
<td>1643 types</td>
</tr>
</tbody>
</table>

Table 5.3: Overview over different versions of Bible corpus used in the development and evaluation of the morphological analyzer

frequency items. This distinction becomes apparent only if the results of the word list are compared with the results of the running test. If the results of the running text are better than the results of the word list, this means that the morphological analyzer does well on higher frequency items.

Such tests on running text and on word lists can be used to optimize the analyzer. For the development of the Murrinh-Patha morphological analyzer, the results of applying the analyzer to the word list and the running text of the development corpus were analyzed carefully.

The development of the morphological analyzer started out with a close modeling of the well-documented facts of the Murrinh-Patha morphology. This included the inflection of nominals with case and discourse markers as well as the implementation of the complex verbal template which required the implementation of the various dependencies between the morphemes with flag diacritics as described in the previous section.

This morphological analyzer was then applied to the development corpus described above. The interpretation of the output led to the addition of new lexical items to the analyzer. However, it should be noted that the output of the morphological analyzer at the development stage could not be checked in fieldwork, and the translation that is provided in the bible corpus is very free. Consequently, the interpretation of the output is quite difficult, and additions to the morphological analyzer have been made only very tentatively.

For this reason, mainly higher frequency items, borrowed nouns and spelling alternatives whose meaning could be deduced from the translation were added. For example, 20 Murrinh-Patha nouns were added. These comprise spelling alternatives or nouns which did not have a lexical entry in Blythe’s toolbox dictionary or in Street (1989), but which could be found in the resources nonetheless, e.g. in example sentences for other lexical entries. Spelling alternatives have been added when the content could verify that indeed the same word was meant. Some examples for spelling alternatives are *kulututuk* in the corpus for which *kurlurnturtuk* ‘dove’ could be found in Street (1989) or *purrkpurrk* in the corpus corresponding to *purrpurrk* ‘small and numerous’ in Street (1989). For borrowed nouns, 201 new lexical
items were added, especially names of people and places, animal names and a couple of words for concepts and objects which seem to not have a corresponding word in Murrinh-Patha, e.g., basket, olive, birthday etc.

After the inspection of the development corpus, the morphological analyzer comprises the lexical entries as specified in Table 5.4. The lexicon contains all forms of the 38 classifier stems as well as all forms of the functional morphemes, i.e., number markers, tense markers etc. It comprises 1041 lexical stems, which include simple lexical stems as well as their lexicalized reduplicated versions. With these lexical stems, 2173 classifier plus lexical stem combinations can be formed, i.e., each lexical stem can combine with roughly two classifier stems. Nouns form the other large class, with 1271 different lexical items. Finally, adjectives, adverbs, interjections, interrogatives etc. have also been implemented in the morphological analyzer.

When this morphological analyzer, i.e., with the additions to the lexicon, is applied to the word list of the development corpus, 82.55% of words can be given an analysis. An analysis of the unanalyzable 17.45% of the output shows that this comprises many verbs. If verbs do not receive an analysis by the morphological analyzer, this may be due to two reasons: either the

<table>
<thead>
<tr>
<th>Lexical items</th>
<th>Number of entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>all forms of 38 classifier stems</td>
<td>38</td>
</tr>
<tr>
<td>incorporated body parts</td>
<td></td>
</tr>
<tr>
<td>incorporated adverbs / particles</td>
<td>21</td>
</tr>
<tr>
<td>lexical stems</td>
<td>1041</td>
</tr>
<tr>
<td>classifier plus lexical stem combinations</td>
<td>2173</td>
</tr>
<tr>
<td>nouns</td>
<td>1273</td>
</tr>
<tr>
<td>noun class markers</td>
<td>10</td>
</tr>
<tr>
<td>pronouns</td>
<td>44</td>
</tr>
<tr>
<td>borrowed nouns</td>
<td>201</td>
</tr>
<tr>
<td>adjectives</td>
<td>175</td>
</tr>
<tr>
<td>adverbs</td>
<td>44</td>
</tr>
<tr>
<td>interjections</td>
<td>55</td>
</tr>
<tr>
<td>interrogatives</td>
<td>23</td>
</tr>
<tr>
<td>demonstratives</td>
<td>10</td>
</tr>
<tr>
<td>non-conjugated verbs</td>
<td>11</td>
</tr>
<tr>
<td>numerals</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.4: Number of lexical entries for the Murrinh-Patha morphological analyzer. Top: different morphemes of the verb; Bottom: stems for other parts of speech.
combination of classifier and lexical stems is unknown or the lexical stem is
unknown.

As can be seen in Table 5.4, 2173 classifier and lexical stem combinations,
i.e., 2173 verbs, have been implemented in the morphological analyzer. It
is very likely that more combinations of lexical and classifier stems exist
and that these have not been listed in the database so far. Similarly, it is
quite likely that other lexical stems exist which are not listed in the
database so far. However, it is desirable that these combinations receive a
morphological analysis as well.

However, as these new combinations have not been documented so far,
they should be specially marked in the output of the morphological ana-
lyzer to distinguish them from the combinations that were known from the
database. This can be accomplished by making use of a stepwise lookup
strategy. A stepwise lookup strategy can be used whenever one analysis is
given priority over another analysis. For this purpose, a morphological ana-
lyzer uses different lookup strategies to analyze the input. The input is first
analyzed by a first strategy of the morphological analyzer. If this strategy
does not provide an analysis for the input, the second strategy is used etc.

Such a stepwise lookup strategy is for example used in a morphological
analyzer for Basque (Alegria et al. 1996). In this analyzer, the first lookup
strategy analyses the standard language. Only if the first lookup strategy
does not provide an analysis, the input is passed on to the second lookup
strategy which analyses linguistic variants such as dialectal constructions.
As a third lookup strategy, Alegria et al. (1996) propose an analysis for
words that are not covered by the lexicon.

For the Murrinh-Patha morphological analyzer, a similar stepwise lookup
strategy is used. The first strategy consists of the morphological analyzer as
described so far. The second lookup strategy covers the cases of unknown
combinations of classifier and lexical stems. This second lookup strategy
only applies if the first strategy cannot provide an analysis. In this way,
priority is given to already known classifier and lexical stem combinations.
The second lookup strategy generates candidates for new classifier and lex-
ical stem combinations only. Their meaning has to be established though
additional analysis, for example by looking at the translation of the corpus
or by additional fieldwork. The following section discusses the interpretation
of the new classifier and lexical stem combinations in detail.

For the implementation of this second lookup strategy, the same finite
state network is used as for the first lookup strategy. The only difference is
that the flag diacritics which model the dependencies between the classifier
and lexical stems have been left out. This means that the network accepts
any combination of classifier and lexical stem. With this second lookup
strategy, 4.28% of the development corpus word list could be analyzed.

As was pointed out above, Murrinh-Patha verbs could also be unanalyz-
able for the computational implementation because the lexical stem involved
is not listed in the lexicon. As is listed in Table 5.4, 1041 lexical stems could be extracted from the database. Lexical stems are the only large class of morphemes that constitute verbs. Classifier stems as well as the other morphemes such as tense and number markers, etc., all come from small, closed classes of morphemes. If a verb can thus not be given an analysis with the first two lookup strategies, it is most likely that the lexical stem is unknown. As with the unknown classifier and lexical stem combinations discussed above, these verbs should receive a morphological analysis. For this purpose, a morphological guesser for lexical stems is incorporated into the system as a third lookup strategy. The morphological guesser enables the network to guess the form of a lexical stem without the need of having the lexical stem in the lexicon.

The guesser for Murrinh-Patha lexical stems has been implemented as described by Beesley & Karttunen (2003) for guessers in general. In a given verb, the classifier stem and the other morphemes, for example tense and number markers, are identified just as in a normal finite-state network. What is left if these morphemes are identified can then be considered the candidate for the lexical stem. (5.22) provides an example. *perremka* can be identified as the third person dual, non-future form of the classifier stem poke:rr(21) while *-neme* is the paucal male number marker. The guessed lexical stem in this case is *wirnturt*.7

\[
\text{(5.22) perremka-wirnturt-neme} \quad \text{3duS.poke:rr(21).nfut-lexical.stem-pauc.m}
\]

In the implementation of the guesser, the LEXC lexicon for lexical stems includes a placeholder entry which is then replaced in a network by all phonologically possible lexical stems. As an analysis of the database used shows, lexical stems need to have at least one syllable and they usually begin with a consonant. However, some lexical stems beginning with the vowel ‘a’ are also attested. These constraints have been integrated into the morphological guesser.

The guessed lexical stems may also undergo phonological changes. This means that different guesses may be possible for one input string. For example, if the lexical stem *ngkardu* ‘see’ were unknown, a possible guess for the input string *bamkardu* would be both, *kardu* and *ngkardu* as lexical stems in combination with the classifier stem 13.

Because of the various guesses due to the phonological rules, the morphological analyzer has to have the second and third lookup strategies as separate steps. The guesser could also cover the known lexical stems in unknown combinations, but due to the phonological rules, the guesser produces

---

7 As is discussed in Section 5.4.2, *wirnturt* is just a spelling alternative to the lexical stem *wirndurt* that is listed in Street (1989).
Figure 5.1: Stepwise lookup strategy for the Murrinh-Patha morphological analyzer for the development corpus.

various possible analyses. Separating the two steps by first relaxing the constraints on the classifier and lexical stem combinations and then guessing unknown lexical stems gives priority to already existing lexical stems. It so constrains the amount of possible analyses for the input strings with a known lexical stem.

A morphological guesser only makes sense for lexical items which obligatorily combine with other morphemes as these morphemes serve as a restrictor for the various guesses. It is therefore helpful for the guesser if all the dependencies between the morphemes are implemented for the guesser as well. For example, in (5.22), the lexical stem can be guessed as wirnturt as there is a restriction that -neme can only attach to dual classifier stems. The guessed lexical stem cannot be kawirnturt, for example, although a classifier stem perrem exists, which is, however, the plural form of the classifier stem POKE:RR(21).

The system is constrained to detecting more Murrinh-Patha verbs. No guesser has been implemented for the other parts of speech as these do not have to be inflected. Nouns offer themselves as candidates for guessers cross-linguistically as nouns usually form a large open class. It is therefore desirable to have a morphological guesser for nouns. However, nouns in Murrinh-Patha do not have to be case-marked or carry other inflection. For this reason, a guesser for nouns is not helpful as all input strings could then be analyzed as nouns, which is not desirable. The third lookup strategy thus only comprises a guesser for lexical stems.

The stepwise lookup strategy as described so far is illustrated in Figure
Table 5.5: Output of the morphological analyzer with three strategies for the development corpus word lists: strategy 1 involves full constraints, strategy 2 has relaxed constraints on classifier and lexical stems and strategy 3 guesses unknown lexical stems.

5.1. The lookup strategy makes the morphological analyzer robust without the danger of overgenerating. Thus, if the system is used to analyze a corpus, the system can detect new combinations of classifier plus lexical stem combinations and new lexical stems.

The results of the morphological analysis of the word list of the development corpus with the described stepwise morphological analyzer are presented in Table 5.5. As can be seen in the table, the overall coverage of the morphological analyzer is already very satisfying, only 0.98% of the words could not be given an analysis. However, the morphological guesser applies to 12.19% of the words, which is quite high.

When looking at the output of the morphological guesser in more detail, it becomes apparent that the morphological guesser provides guesses for lexical stems in verbs which could also be analyzed differently. This is the case for some morpheme orderings which have not been described before, or which were described as ungrammatical. For example, in (5.23), the adverb *deyida* ‘again’ occurs before the tense marker *-dha* while following the verbal template, it should only be able to attach after the tense marker. As a result, the lexical stem guessed by the morphological analyzer is *ngerrendeyida*.

(5.23) kardi-ngerren-*deyida*-dha
3SGS.BE(4).PIMPF-talk-again-PIMPF

To be able to provide a more coherent analysis of such unexpected morpheme orderings, another strategy has been added to the stepwise morphological analyzer. These new morpheme orderings, as well as some relaxed constraints on the morpheme cooccurrences, have been given a separate lookup step and not been included in the basic morphological analyzer to be able
to consider these cases separately. In the following, the morpheme orderings and relaxed constraints for this new lookup strategy are described briefly.

As has been described above already, the lookup strategy 2 relaxes the constraints on the combination of classifier and lexical stems to detect new combinations of classifier and lexical stems. In the new lookup strategy, the constraints on the tense markers and on the number markers have been relaxed. For tense markers, Nordlinger & Caudal (2012) describe the known combinations of classifier stem inflection and separate tense markers. However, the corpus contains some instances of verbs which do not follow the description in Nordlinger & Caudal (2012). An example is provided in (5.24).

Both the past imperfective and the past irrealis form of the classifier stem require the additional tense marker -dha, which is missing in this case.

(5.24) bina-na-yepup
3SGS.16.PIMPF/PSTIRR-3SG.BEN-listen

Similarly, the corpus contains instances of the number markers -nintha, -ngintha, -neme and -ngime which are not licensed by the classifier stem or the object markers. As has been discussed above, the constraints on these number markers are quite complex. They can be licensed by the classifier stem (-nintha/-ngintha can only occur with singular classifier stems and -neme/-ngime can only occur with the dual classifier stem) or by certain direct or indirect object markers. In (5.25), the dual non-sibling female marker -ngintha is not licensed because the classifier stem is either dual or plural and the second person singular indirect object marker -mpa also does not combine with an additional number marker normally.

(5.25) ngira-mpa-winhadhath-tha-ngintha
1DU/PLS.WATCH(28).PIMPF-2SG.BEN-look.for-PIMPF-DU.F

To be able to provide these and similar ungrammatical or at least undescribed combinations of morphemes with an analysis, the constraints on the number and tense markers have been relaxed in the new lookup strategy.

Besides relaxed constraints, some undescribed morpheme orderings are also allowed in this lookup strategy. Firstly, the number markers, most often -neme, occur at the end of the verb, even after discourse marking etc.\(^8\)

In (5.26), the number marker -neme occurs after the emphatic discourse marker -wa. That the number marker occurred at the end of the verb was quite common in the corpus.

(5.26) pume-berti-dha-mana-wa-neme
3DU/PLS.HANDS(8).PIMPF-take-PIMPF-only-DM-PAUC.M

\(^8\)The discourse markers were not part of the verbal template presented in Figure 5.1 as it seems that they can attach to the end of almost every word.
Secondly, the incorporated adverb *deyida* ‘again’ occurred in unexpected slots, before the tense marker, as was already shown in (5.23). In the new lookup strategy, an additional slot for *deyida* between the lexical stem and the tense marker has been added.

Thirdly, verbs seem to be able to take case marking, most often the case marker *-re/-te*. Most often these case markers occur before the discourse markers such as in (5.27).

(5.27)  
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>nuru-rdurr-nu-re-ka</td>
</tr>
<tr>
<td></td>
<td>2PLS.GO(6).FUTIRR-leave-FUT-ERG/INSTR-DM</td>
</tr>
<tr>
<td>b.</td>
<td>nuru-lili-nu-re-yu</td>
</tr>
<tr>
<td></td>
<td>2PLS.GO(6).FUTIRR-walk-FUT-ERG/INSTR-DM</td>
</tr>
</tbody>
</table>

However, examples in which the case marker attaches after the discourse marker can be found as well. This ordering is also found for other parts of speech. For example, in (5.28), the pronoun *nukunu* occurs with the discourse marker *-wa* and the dative marker *-nu*.

(5.28)  
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nukunu-wa-nu</td>
</tr>
<tr>
<td></td>
<td>3SG.M-DM-DAT</td>
</tr>
</tbody>
</table>

Finally, the case markers *-nu* and *-re/-te* seem to be able to occur together. Examples from a wide range of parts of speech can be found. For example, in (5.29), the demonstrative *nhini* occurs with both case markers, while in (5.29b) an adjective and in (5.29c) an adverb is involved.

(5.29)  
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a.</td>
<td>nhini-nu-re</td>
</tr>
<tr>
<td></td>
<td>demonstrative-DAT-ERG/INSTR</td>
</tr>
<tr>
<td>b.</td>
<td>manta-re-nu</td>
</tr>
<tr>
<td></td>
<td>near-ERG/INSTR-DAT</td>
</tr>
<tr>
<td>c.</td>
<td>nakurl-te-nu</td>
</tr>
<tr>
<td></td>
<td>later-ERG/INSTR-DAT</td>
</tr>
</tbody>
</table>

The morphological analyzer with this additional lookup strategy is displayed in Figure 5.2. The new lookup strategy is added before the guesser as new third strategy. The guesser is now the fourth lookup strategy.

This morphological analyzer with the four lookup strategies then yields the output provided in Table 5.6 for the development corpus word list. As can be seen, the new strategy accounts for 5.55% of all words, and makes the output of the morphological guesser considerably smaller. Additionally, the new strategy also accounts for some words which could not be analyzed before at all.

A final note on the morphological guesser: the guesser used in the lookup strategy with the new ordering possibilities is not altered, it is still built on
Figure 5.2: Final stepwise lookup strategy for the Murrinh-Patha morphological analyzer.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Words</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy 1</td>
<td>4644 words</td>
<td>82.55 %</td>
</tr>
<tr>
<td>Strategy 2</td>
<td>241 words</td>
<td>4.28 %</td>
</tr>
<tr>
<td>Strategy 3</td>
<td>312 words</td>
<td>5.55 %</td>
</tr>
<tr>
<td>Strategy 4</td>
<td>384 words</td>
<td>6.83 %</td>
</tr>
<tr>
<td>Not found</td>
<td>45 words</td>
<td>0.80 %</td>
</tr>
</tbody>
</table>

Corpus size: 5626 words
Execution time: 1 sec
Speed: 5626 words/sec

Table 5.6: Output of the final morphological analyzer with four lookup strategies for the development corpus word list. The strategy 3 is newly introduced and accounts for unexpected morpheme orderings. Strategy 4 is now the morphological guesser.
### Table 5.7: Output for the morphological analyzer for the development corpus running text, i.e., with word frequencies taken into account.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Words</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy 1</td>
<td>58486 words</td>
<td>97.35%</td>
</tr>
<tr>
<td>Strategy 2</td>
<td>334 words</td>
<td>0.56%</td>
</tr>
<tr>
<td>Strategy 3</td>
<td>453 words</td>
<td>0.75%</td>
</tr>
<tr>
<td>Strategy 4</td>
<td>568 words</td>
<td>0.95%</td>
</tr>
<tr>
<td>Not found</td>
<td>238 words</td>
<td>0.40%</td>
</tr>
<tr>
<td>Corpus size</td>
<td>60079 words</td>
<td></td>
</tr>
<tr>
<td>Execution time</td>
<td>2 sec</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>30039 words/sec</td>
<td></td>
</tr>
</tbody>
</table>

When this morphological analyzer is used for lookup of the running text of the development corpus, i.e., when the word frequencies are taken into account, one finds that the results are very satisfying, as can be seen in Table 5.7. Already the first step finds analyses for 97% of all words, and only 0.4% of words cannot be analyzed. In comparison with the results for the word list this shows that the morphological analyzer is very good for the frequent items, i.e., that the words which cannot be analyzed with strategy 1 are all rather infrequent. This is further support for keeping the strategies apart. The output of strategy 2, 3 and 4 is separate from the output of the first strategy and can be checked further, for example in future fieldwork.

The results of the development corpus can also be compared to the results of the test corpus which are given in Table 5.8 and Table 5.9. Table 5.8 shows the results of the test corpus word list. It can be seen that the results for the first lookup strategy are even higher for the test corpus than for the development corpus (86% vs. 83%). The results for the second and third lookup strategy do not differ greatly while the result of the guesser (strategy

---

9Seiss (2012b) proposes two guessers in which the later builds on relaxed constraints of the tense markers. However, this guesser does not account for many analyses.
strategy 1: 1417 words 86.24%
strategy 2: 63 words 3.83%
strategy 3: 81 words 4.93%
strategy 4: 50 words 3.04%
not found: 32 words 1.95%
corpus size: 1643 words
execution time: 0 sec
speed: 1643 words/sec

Table 5.8: Output for the morphological analyzer for the test corpus word list.

4) is lower for the test corpus compared to the development corpus. In total, 1.95% of words could not be given an analysis for the test corpus in contrast to only 0.8% of words which could not be analyzed in the development corpus. This is mainly due to the relatively large number of borrowed nouns that were part of the test corpus but which never turned up in the development corpus. Twenty new borrowed nouns could be detected in the unanalyzable output of the test corpus which were not part of the development corpus. If one adds these twenty nouns to the analyzer, only twelve words would remain unanalyzable, which would result in 0.7% of unanalyzable output for the test corpus, which is very similar to the output of the development corpus.

This shows that the development corpus was probably too small to really cover a satisfying amount of lexical items which are able to be borrowed into Murrinh-Patha from English. However, the results of the other lookup strategies and the rest of the unanalyzable output show that the corpus is large enough to get an insight into the complexities of Murrinh-Patha words. The similarity of the output for the test corpus words with word frequencies taken into account, displayed in Table 5.9, to the development corpus words confirms this view.

To sum up, a stepwise lookup strategy was developed for the Murrinh-Patha morphological analyzer by carefully analyzing the various stages of the development corpus output. This yielded a morphological analyzer that has satisfying coverage for both the word list and the running text of the test corpus. However, the results presented so far only indicate whether a morphological decomposition can be found for a given string of words and so measures pure coverage. This is known as negative testing. Of course,
it would be desirable to also have positive testing, i.e., to test whether the analyses which were found were correct.

Unfortunately, positive testing, in contrast to negative testing, cannot be carried out systematically easily for Murrinh-Patha. Positive testing requires a Gold Standard, but the resources for this are not available right now, as creating a Gold Standard would require extensive fieldwork on Murrinh-Patha.

Nevertheless, the results of the morphological analyzer can be evaluated qualitatively. This is carried out in the following section for the output of the lookup strategies 2 and 4. These lookup strategies were designed to find new verb candidates. Strategy 2 was designed to find new classifier plus lexical stem combinations and strategy 4, the morphological guesser, finds new candidates for lexical stems. The following section evaluates what can be deduced from this output even without fieldwork by just using the available resources and the English translations of the bible corpus.

### 5.4 New verbal candidates from the corpus

The previous section described the development of a stepwise morphological analyzer for Murrinh-Patha. It argued for a morphological analyzer with four different lookup strategies. The first lookup strategy covers all previously described phenomena and lexical items of Murrinh-Patha. The second strategy relaxes the constraints on the classifier and lexical stem combinations and so allows to detect new combinations. The third strategy allows for so far undescribed morpheme orderings and dependencies while the fourth strategy is a morphological guesser for lexical stems. The second and fourth lookup strategies both return new candidates for Murrinh-Patha verbs. This
section discusses the new verb candidates and how these candidates can be interpreted.

For the identification of new verbal candidates, both morphological as well as syntactic clues can be used. The morphological analyzer is only concerned with morphological clues. For the corpus analysis presented in this section, no automatic analysis of the syntactic clues has been carried out as a computational syntactic analysis would introduce more ambiguities than it could resolve. This is discussed in Chapter 6 in more detail.

However, the new verbal candidates rendered by the morphological analyzer have been inspected manually, and this manual inspection also took syntactic clues into account. For many candidates, the morphological clues alone were enough to establish their status as verbs. This was especially the case if the verb did not only consist of a classifier and lexical stem, but if additional morphemes were part of the verbal template that adhered to the constraints described in Section 5.1.

(5.30) provides examples for these definite morphological clues. In (5.30a), the future tense marker -nu agrees with the tense marking on the classifier stem. This tense marking in addition to the RR marker -nu also specifies that the classifier stem can only be interpreted as the RR classifier stem HANDS:RR(10) and not its non-RR classifier stems HANDS(8) or SNATCH(9) whose past imperfective form is also me. Similarly, in (5.30b), the discontinuous benefactive marker as well as the incorporated body part show that this is indeed an instance of a verb.

(5.30) a. me-nu-yibimu-nu
   3SGS.HANDS:RR(10).FUT-RR-save.life-FUT

b. wurdan-pirru-me-riyith-neme
   3SGS.SHOVE(29).NFUT-3BEN-IBP-explain-PAUC.M

If the verbal candidate is analyzed as consisting only of a classifier stem and lexical stem, the morphological clues may not be enough to determine its status as a verb completely. In this case, syntactic clues may help identify whether the candidate is indeed a verb. If there is no other verb in the clause, the verbal candidate is very likely a verb. Additionally, if the verbal candidate is followed by an intransitive posture or motion classifier stem that agrees with the verbal candidate in subject number, person and tense, the verbal candidate is most likely a verb and the intransitive posture and motion classifier stem a clitic as described in Nordlinger & Caudal (2012).

An example is provided in (5.31) in which the classifier stem HANDS:RR(10) combines with the lexical stem wiye ‘spoil’. The classifier stem GO(6) is cliticized on the verb candidate and agrees with it in subject person, number and tense. The syntactic clues therefore confirm that the verbal candidate memwiye is indeed a verb.
Such clues are used in the following subsections to decide whether a verbal candidate is indeed a verb. To establish the meaning of the new verbal candidate, the English translations offered by the corpus have been investigated. However, the translations are unfortunately very free. For most of the sentences, a very long Murrinh-Patha paragraph corresponds to a short sentence in English. This free translation makes it difficult to guess the meaning of the new Murrinh-Patha verbs. As such, the meaning of most of the newly found verb candidates can only be determined in future fieldwork. However, for some of the new verb candidates, especially the more frequent ones, a meaning can also be deduced from the translation.

More generally, the new verb candidates can be grouped into various subgroups to make an interpretation easier. The remainder of this section deals with these subgroups and discusses some of the candidates for which a meaning can be suggested even from the corpus. Subsection 5.4.1 first discusses the output of strategy 2, i.e., new combinations of already known lexical stems. In subsection 5.4.2, new candidates for lexical stems are discussed.

5.4.1 New combinations of classifier and lexical stems

For the new combinations of classifier and lexical stems, different subgroups can be formed based on the expected possibility of the lexical stems to combine with certain classifier stems. This section discusses three such subgroups, starting with the subgroup of combinations which are expected based on their behavior with transitive classifier stems and their respective reflexive/reciprocal versions. Another subgroup can be formed based on the behavior of lexical stems with motion and posture classifier stems. Finally, the section also discusses combinations which are not expected based on the first two criteria but which occur more than twice in the corpus.

Correspondences between transitive classifier stems and their reflexive/ reciprocal versions can be used to interpret newly found classifier and lexical stem combinations. As has been laid out in Chapter 3, the Murrinh-Patha classifier stems can be divided into intransitive, transitive and reflexive/reciprocal classifier stems. As has been discussed by Nordlinger (2008), the reflexive/reciprocal classifier stems usually correspond to one or two transitive classifier stems. Table 5.10 provides the correspondences established by Nordlinger (2008).
These correspondences can be used to interpret the newly found classifier plus lexical stem combinations, i.e., if a combination of a reflexive/reciprocal classifier stem with a lexical stem is detected for which the corresponding transitive classifier stem combines with the same lexical stem, this combination is not surprising and can be interpreted more easily. Similarly, a new combination of a transitive classifier stem with a lexical stem for which the reflexive/reciprocal classifier stem is known to combine with the lexical stem is not surprising either.

Table 5.11 provides the new combinations of classifier and lexical stems for which either the transitive or the reflexive/reciprocal version is known. A rather clear example is provided by the lexical stem *yibirnu*. In the database, *yibirnu* is listed with the classifier stem HANDS(8) to mean ‘save somebody’s life’. In the corpus, it occurs twice with the classifier stem HANDS:RR(10). One of the occurrences is given in (5.32). The translation verifies that *yibirnu* in combination with the classifier stem HANDS:RR(10) means something like ‘save own’s life, hold on to own’s life’.10

(5.32) **Murrinh-Patha:** I kardu nangkal ngatha ngay paningipartnu nhini kardu ngamere kathu mere ngarra thu kubatnu, nhini ngay ngamam-ka nukunu nhini-ka pupup da matha-nu nhini-yu, i mere ngarra me-nu-yibirnu (3SGS.HANDS:RR(19).FUTIRR-RR-save.life) nukun mange nukunu ngatha-yu, wurda. Mu kardu nangkal ngatha punymyekumnu-ka, i ngay da matha kumparra bungipaknu kurr, kardu nhini-ka bere matha wangu birnu paninu.

**English:** Whoever tries to hold on to his life will give up true life, Whoever gives up his life for me will hold on to true life.

---

10In the corpus examples, the context of the verb occurrence as well as the English translation is provided. To make the examples more understandable, a gloss is provided for the verb in parentheses.
<table>
<thead>
<tr>
<th>Lexical stem</th>
<th>new CS</th>
<th>known CS</th>
<th>#</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>art</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>6</td>
<td>numenurtnu</td>
</tr>
<tr>
<td>riyith</td>
<td>SHOVE(29)</td>
<td>SHOVE:RR(30)</td>
<td>6</td>
<td>wurdanpirrunerythneme</td>
</tr>
<tr>
<td>mut</td>
<td>POKE:RR(21)</td>
<td>POKE(19)</td>
<td>5</td>
<td>nenumut</td>
</tr>
<tr>
<td>yibirnu</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>5</td>
<td>menuyibirnu</td>
</tr>
<tr>
<td>(marda)puth</td>
<td>SHOVE:RR(30)</td>
<td>SHOVE(29)</td>
<td>2</td>
<td>wurdammardaputh</td>
</tr>
<tr>
<td>rart</td>
<td>HANDS:RR(10)</td>
<td>SNATCH(9)</td>
<td>2</td>
<td>numerart</td>
</tr>
<tr>
<td>ma</td>
<td>POKE(19)</td>
<td>POKE:RR(21)</td>
<td>1</td>
<td>daninmadha</td>
</tr>
<tr>
<td>ngkarl</td>
<td>SHOVE:RR(30)</td>
<td>SHOVE(29)</td>
<td>1</td>
<td>thurdingangkarl</td>
</tr>
<tr>
<td>ngkarnu</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>mem-ninthang-karnu</td>
</tr>
<tr>
<td>ngurruduk</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>memmu-mangurruk-duk</td>
</tr>
<tr>
<td>nirn</td>
<td>BASH:RR(15)</td>
<td>BASH(14)</td>
<td>1</td>
<td>bem-pirrang-kanirn</td>
</tr>
<tr>
<td>pe</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>nemnupe-wa</td>
</tr>
<tr>
<td>rdarrithak</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>merdarrithak-mana-wa</td>
</tr>
<tr>
<td>rnama</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>memnumarna</td>
</tr>
<tr>
<td>rtal</td>
<td>BASH:RR(15)</td>
<td>BASH(14)</td>
<td>1</td>
<td>nubemartal-wa</td>
</tr>
<tr>
<td>rte</td>
<td>SLASH:RR(24)</td>
<td>SLASH(23)</td>
<td>1</td>
<td>tham-punku-mardarte-neme</td>
</tr>
<tr>
<td>wath</td>
<td>HANDS(8) or SNATCH(9)</td>
<td>HANDS:RR(10)</td>
<td>1</td>
<td>mawathnu-yu</td>
</tr>
<tr>
<td>wiye</td>
<td>HANDS:RR(10)</td>
<td>HANDS(8)</td>
<td>1</td>
<td>mempiye</td>
</tr>
<tr>
<td>yethith</td>
<td>SHOVE:RR(30)</td>
<td>SHOVE(29)</td>
<td>1</td>
<td>purdi-nhi-yethith-nungime</td>
</tr>
</tbody>
</table>

Table 5.11: New classifier and lexical stem combinations: reflexive/reciprocal or non-reflexive/reciprocal (i.e., transitive) versions of already known transitive or reflexive/reciprocal combinations and their number of occurrences in the corpus (#).
Note that the translation in (5.32) is quite free and the Murrinh-Patha paragraph is much longer than the English paragraph. Thus, without the knowledge that *yibirnu* has the meaning ‘save someone’s life’ with the classifier stem HANDS(8), the meaning of *yibirnu* in combination with the classifier stem HANDS:RR(10) probably could not have been established from the translation.

In this example, the relationship between the transitive and the reflexive/reciprocal classifier stems is regular, i.e., the complex predicate formed with the reflexive/reciprocal classifier stem is the reciprocal version of the complex predicate formed with the corresponding transitive classifier stem. This is not always the case, however. The reflexive/reciprocal classifier stems are also sometimes used to denote other intransitive events (as has been discussed in Section 3.6) and rarely even transitive events. However, even in such cases it may help to look at the corresponding transitive classifier stem.

An example for such a case is provided by the lexical stem *riyith*. In the database, this lexical stem is listed with the classifier stem SHOVE:RR(30), meaning ‘explain’. It occurs six times with the classifier stem SHOVE(29) in the corpus and it seems to mean ‘explain’ in these cases, too. This can be verified in this case because the expression occurs in titles which are closely translated and also because the combination occurs considerably often in the corpus. Examples are provided in (5.33).

(5.33)  

a. **Murrinh-Patha**: Jesus *Wurdan-pirru-me-riyith-neme*  
(3SGS.SHOVE(29).NFUT-3.BEN-IBP-explain-PAUC.M) Murrinh Ngarra Mi Wit i Parnu Wiye  
**English**: Jesus Explains About the Wheat and Weeds

b. **Murrinh-Patha**: Jesus *Wurdan-pirru-me-riyith-neme*  
(3SGS.SHOVE(29).NFUT-3.BEN-IBP-explain-PAUC.M) Murrinh Ngarra Yungkamnumtha  
**English**: Jesus Explains the Seed Story

c. **Murrinh-Patha**: Ya pule!  
Thurdu-ngarru-me-riyith-neme-nu  
**English**: Explain to us the meaning of the story about the weeds in the field.

That the lexical stem *riyith* means ‘explain’ both with SHOVE(29) and SHOVE:RR(30) of course raises the question whether there is a meaning distinction or a distinction in the syntactic behavior between the two constructions. What should be noted is that *riyith* plus SHOVE(29) always occurs with a benefactive marker in the corpus whereas the examples given in the database do not involve benefactive markers.
Of course, the database could also be incorrect in this case. This could have happened because the classifier paradigms for shove(29) and shove:rr(30) are quite similar, e.g., wurdan (29) vs. wurdam (30) and ngurdu (29) vs. ngurdi (30). Evidence that the database might be erroneous in this case comes from the fact that the corpus does not contain an instance of riyith plus classifier stem shove:rr(30). This could be tested in future fieldwork.

A third example that is discussed from this set of newly found combinations is the lexical stem (marda)puth as it again shows the possibilities and limitations of the corpus study. Although the combination of this lexical stem with the classifier stem shove:rr(30) is only found twice in the corpus, its meaning can be determined quite surely as ‘take a deep breathe, sigh deeply’. This is possible because both occurrences in the text are quite closely translated, so that the meaning of the verb can be guessed. The contexts for the occurrences are provided in (5.34).

(5.34) a. **Murrinh-Patha:** Bere Jesus-ka wurdam-mardaputh (3sgS.shove:rr(30).nfut-sigh) i mampirruneme,...
   **English:** Jesus sighed deeply. He said,...

b. **Murrinh-Patha:** I nhini thangunu Jesus-ka kangkarl mamka i ngirkirt ngadha wurdam-mardaputh (3sgS.shove:rr(30).nfut-sigh).
   **English:** Jesus looked up to heaven and took a deep breath.

However, for the lexical stem (marda)puth, one question cannot be answered completely by the corpus study. As has been indicated by the parentheses, it is not entirely clear whether the form of the lexical stem is puth which co-occurs with the incorporated body part marda by coincidence, or whether the lexical stem should be considered to be mardaputh, with a lexicalized incorporated body part.

That lexical stems are formed with lexicalized incorporated body parts is quite common in Murrinh-Patha. Forshaw (2011) treats the incorporation of body parts into Murrinh-Patha verbs as a continuum, i.e., he suggests that some incorporations may be fully productive while others are highly lexicalized.

The lexical stem puth is listed as separate lexical item in the database as well as with a range of incorporated body parts, forming lexicalized lexical stems such as maputh (with the incorporated body part ma ‘hand’) meaning ‘disturb from rest’ in combination with the classifier stem hands(8) or ngkaputh (with the incorporated body part ngka ‘eye’) meaning ‘fight with people’ in combination with the classifier stem shove:rr(30). The deduced meaning of mardaputh ‘take a deep breath, sigh deeply’ cannot be easily linked to the meaning of these lexical stems. Moreover, the deduced meaning can also not be linked to the meaning of puth with shove(29)
'take something off the fire'. This speaks in favor of a lexicalized incorporated body part, i.e., mardaputh should be considered the lexical stem in this case, as the meaning of mardaputh cannot be deduced from the meaning of marda and puth. In this case then, it is not helpful to know that puth is known in the database with SHOVE(29), as the lexical stem in question is most probably mardaputh.

So far, the section discussed how classifier and lexical stem combinations can be interpreted for which a corresponding combination with a transitive or reflexive/reciprocal classifier stem with the same lexical stem exists. For most of the combinations, the knowledge of the corresponding combination helped with interpreting and deducing the meaning of the new combination.

Similar results can also be found for lexical stems which combine with the classifier stems 1 to 6 which are classifier stems of posture and motion. For many lexical stems, the database actually lists all classifier stems 1 to 6 as possible combinations. However, some lexical stems are only listed in the database with one or a subset of the six posture and motion classifier stems. If these lexical stems are found in the corpus with an unlisted posture or motion classifier stem, the knowledge of the combination of this lexical stem with other posture or motion classifier stems can help determine the meaning of the new combination.

An overview over the newly found combinations of posture and motion classifier stems and lexical stems can be found in Table 5.12. As an example, the lexical stem pup will be discussed briefly. The lexical stem pup will be discussed briefly. The lexical stem pup is listed in the database with the classifier stem lie(2) with the meaning 'die', or with the classifier stems stand(3) and be(4) with the meaning 'sit down, live in'. In the corpus, the lexical stem pup occurs 5 times with the classifier stem go(6). In some of the occurrences, it can be deduced that the combination of the lexical stem pup with the classifier stem go(6) can be used to denote the meaning 'live in' as well. (5.35) provides examples of the occurrences of pup in the corpus.

(5.35) Murrinh-Patha: Nga, ku were pulangarr puppan-nu-pup-ka
(3PLS.GO(6).NFUT-RR-live.in-FOC) ngarra da weyi-wa, i ku murriirbe-ka ngarra da ku dirri puppan-nu-pup.
(3PLS.GO(6).NFUT-RR-live.in)

English: The foxes have holes to live in. The birds have nests to live in.

When looking at Table 5.12, one difficulty of the corpus study approach becomes obvious. In the paradigms for the classifier stems, a considerable degree of syncretism can be found. That means that a surface form may be analyzed as many different classifier stems. Additionally, the incorporated body parts and the object markers which attach between the classifier and the lexical stem can also result in different possible analyses for one surface
<table>
<thead>
<tr>
<th>Lexical stem</th>
<th>new CS</th>
<th>known CS</th>
<th>#</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>pup</td>
<td>GO(6)</td>
<td>LIE(2), STAND(3), BE(4)</td>
<td>5</td>
<td>name-pup-nu-ngime</td>
</tr>
<tr>
<td>wuthputh</td>
<td>GO(6)</td>
<td>STAND(3)</td>
<td>5</td>
<td>wurriniwuthpuththa</td>
</tr>
<tr>
<td>wuwup</td>
<td>GO(6) (or SLASH(23))</td>
<td>LIE(2), BE(4)</td>
<td>4</td>
<td>pumpanpuuwup-yu</td>
</tr>
<tr>
<td>wuwup</td>
<td>SIT(1), STAND(3)</td>
<td>LIE(2), BE(4)</td>
<td>2</td>
<td>pinganpuuwuptha</td>
</tr>
<tr>
<td></td>
<td>(or CS 25, WATCH:RR(36))</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rdurdi</td>
<td>STAND(3)</td>
<td>BE(4)</td>
<td>3</td>
<td>pirdurdidha-ya</td>
</tr>
<tr>
<td>rdurdi</td>
<td>GO(6)</td>
<td>BE(4)</td>
<td>1</td>
<td>pirdurdidha-ya</td>
</tr>
<tr>
<td>ngurrkurrk</td>
<td>BE(4)</td>
<td>LIE(2)</td>
<td>3</td>
<td>pardingurrkurrktha-ka</td>
</tr>
<tr>
<td></td>
<td>(or POKE(19), CS 20, TAKE(22))</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rrit</td>
<td>SIT(1)</td>
<td>BE(4), GO(6)</td>
<td>2</td>
<td>dinirrittha</td>
</tr>
<tr>
<td>bat</td>
<td>SIT(1) (or CS 16)</td>
<td>STAND(3), BE(4), 7</td>
<td>1</td>
<td>dimmat</td>
</tr>
<tr>
<td>kutkut</td>
<td>SIT(1), LIE(2) or STAND(3)</td>
<td>BE(4), GO(6)</td>
<td>1</td>
<td>ngarrimkutkut-yu</td>
</tr>
<tr>
<td>mamath</td>
<td>GO(6)</td>
<td>SIT(1), BE(4)</td>
<td>1</td>
<td>wurranmamath</td>
</tr>
<tr>
<td>mum</td>
<td>GO(6)</td>
<td>STAND(3)</td>
<td>1</td>
<td>kananum</td>
</tr>
<tr>
<td></td>
<td>(or POKE(19), CS 20, TAKE(22))</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rartal</td>
<td>GO(6)</td>
<td>SIT(1)</td>
<td>1</td>
<td>wurrirartaldha</td>
</tr>
<tr>
<td>rrurrurr</td>
<td>GO(6)</td>
<td>SIT(1), STAND(3)</td>
<td>1</td>
<td>wurrarnurrrurr</td>
</tr>
<tr>
<td>thudhuth</td>
<td>GO(6)</td>
<td>BE(4)</td>
<td>1</td>
<td>wurrinithudhuththa</td>
</tr>
</tbody>
</table>

Table 5.12: New classifier and lexical stem combinations: Lexical stems already known with other posture and motion classifier stems (CS) and their number of occurrences in the corpus (#).
form. Thus, if a lexical stem only occurs in an unknown combination with a classifier stem once, or always with the same surface form, it is difficult to determine with which classifier stem the lexical stem actually combines.

For example, the surface form pardingurrkurrktha-ka could be a combination of the lexical stem ngurrkurrk with either pardi (a surface form of the classifier stem BE(4)) or a surface form pa (forms of the classifier stems 19, 20 and 21) in combination with the incorporated body part rdi ‘tooth’. In this case, it is more likely that the form involved is pardi, as this involves only one morpheme and not two separate morphemes pa-rdi. Additionally, and more generally, it is more likely that the classifier stem in the new combination is a posture or motion classifier stem if combinations of the same lexical stem with other posture and motion classifier stems exist. For this reason, the posture and motion classifier stems are more likely in these combinations, and the other classifier stems have been put in parentheses in Table 5.12.

Finally, a couple of new classifier plus lexical stem combinations can be extracted which do not belong to the classes discussed above, i.e., the co-occurrence of the lexical stem with a specific classifier stem is not expected based on the similarity of already known combinations. In Table 5.13, all new combinations are displayed which occur at least twice. Of course, for these combinations it is even more difficult to guess the meaning of the new combinations. However, detailed analyses can help determine the meaning of some of the new combinations, especially the more frequent ones. In the following, this will be shown for some exemplary combinations.

The most frequent new combination is the lexical stem (ngka)birl combining with the classifier stems SHOVE(29) and SHOVE:RR(30). The lexical stem birl is listed with the classifier stems HANDS(8) and HANDS:RR(10) to denote the meanings ‘to rub off singed fur/feathers’ and ‘to turn one’s head around to look’ respectively. Additionally, the lexical stem ngkabirl is listed with the classifier stem SLASH(23) and the meaning ‘to kindle a fire by blowing’. None of these meanings seems to fit to the contexts the lexical stem (ngka)birl is used in the corpus. However, ngkabirl is also noted in the database as ‘to come to life’, but this is not in a lexical entry for ngkabirl, but in a lexical entry for the noun and adjective birnu ‘life, alive’ for which ngkabirl is listed as a synonym. That is, this meaning of ngkabirl is hidden in the database but is not in a form in which it is easily accessible and it also does not provide the classifier stems with which this meaning of ngkabirl occurs.

When looking at the occurrences of the lexical stem in the corpus, it becomes evident that this meaning is most probably associated with the lexical stem in combination with the classifier stems SHOVE(29) and SHOVE:RR(30). The verb is used very often in combination with the name Elijah, a prophet who was able to raise the dead. Also, the contexts in (5.36) are so closely translated that it can be assumed that ngkabirl plus the classifier stem
<table>
<thead>
<tr>
<th>Lexical stem</th>
<th>new CS</th>
<th>#</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ngka)birl</td>
<td>SHOVE:RR(30)</td>
<td>19</td>
<td>thurdamkabirl, wurdamkabirl, wurdamkabirl</td>
</tr>
<tr>
<td>(ngka)birl</td>
<td>SHOVE(29)</td>
<td>8</td>
<td>ngurdungkabirl-deyi-da-nu, ngurdungkabirlnu</td>
</tr>
<tr>
<td>(ngka)rdurdu</td>
<td>BASH:RR(15)</td>
<td>20</td>
<td>pubemkardurdu, nubemkardurdu, pubeng-kardurdu-nemenu</td>
</tr>
<tr>
<td>thuk</td>
<td>HANDS(8), SNATCH(9) or HANDS:RR(10)</td>
<td>5</td>
<td>merithuktha</td>
</tr>
<tr>
<td>thuk</td>
<td>HANDS(8)</td>
<td>2</td>
<td>thu-marnitithuk-ngime</td>
</tr>
<tr>
<td>thuk</td>
<td>HANDS:RR(10)</td>
<td>2</td>
<td>pumarnitithuk</td>
</tr>
<tr>
<td>thuk</td>
<td>POKE(19), CS 20 or TAKE(22)</td>
<td>2</td>
<td>dengarru-thuk-tha-neme, nganathuknu</td>
</tr>
<tr>
<td>ngkarr</td>
<td>POKE:RR(21)</td>
<td>6</td>
<td>demnungkarr</td>
</tr>
<tr>
<td>ngkarr</td>
<td>TAKE(22)</td>
<td>3</td>
<td>kaunthinkarr, thanthin-pun-karr, nganthin-ning-karr</td>
</tr>
<tr>
<td>ngkardu</td>
<td>BASH(14)</td>
<td>8</td>
<td>pu-bangam-mang-kardu, bangammangkardu,</td>
</tr>
<tr>
<td>(we)lerr</td>
<td>BASH(14)</td>
<td>7</td>
<td>nubawelerrnenu, nubawelerrdha</td>
</tr>
<tr>
<td>(we)wal</td>
<td>SIT(1)</td>
<td>7</td>
<td>dimpewal, kinhiwewal, ngempewal</td>
</tr>
<tr>
<td>(ye)mit</td>
<td>SHOVE(29)</td>
<td>3</td>
<td>muddu-na-yemit-nu, purdu-nga-yemit-nu, thudu-uyemit</td>
</tr>
<tr>
<td>(ye)mit</td>
<td>SHOVE:RR(30)</td>
<td>2</td>
<td>wurdanayemittha, puddam-na-yemit-ngime</td>
</tr>
<tr>
<td>thi</td>
<td>POKE(19)</td>
<td>5</td>
<td>damna-rithi, danta-rithi, piara-karathi-ka</td>
</tr>
<tr>
<td>terrarr</td>
<td>BREAK(11)</td>
<td>4</td>
<td>munganterrarr, munirerrarrdha</td>
</tr>
<tr>
<td>ngkadhuk</td>
<td>TAKE(22)</td>
<td>3</td>
<td>nanthin-kadhuk, thanthing-kadhuktha</td>
</tr>
<tr>
<td>ngkarda</td>
<td>BASH(14)</td>
<td>3</td>
<td>bangamaka, bangkasadha</td>
</tr>
<tr>
<td>dharlurl</td>
<td>SHOVE:RR(30)</td>
<td>3</td>
<td>muddana-dharlurl-dha-nem, puddana-dharlurl-dha-nem</td>
</tr>
<tr>
<td>werr</td>
<td>HANDS(8)</td>
<td>1</td>
<td>mampunkumardawerneme</td>
</tr>
<tr>
<td>werr</td>
<td>HANDS(8), SNATCH(9) or HANDS:RR(10)</td>
<td>1</td>
<td>menmawerndha</td>
</tr>
<tr>
<td>werr</td>
<td>HANDS:RR(10)</td>
<td>1</td>
<td>nemenu-marda-werr</td>
</tr>
<tr>
<td>ngkayi</td>
<td>GO(6) or SLASH(23)</td>
<td>2</td>
<td>putpan-punkung-kayi-neme</td>
</tr>
<tr>
<td>rartal</td>
<td>BREAK(11)</td>
<td>2</td>
<td>murartal, punungamka-werartal-neme</td>
</tr>
<tr>
<td>warl</td>
<td>SIE(13) or BASH(14)</td>
<td>2</td>
<td>banhirdiwarli, dardiwarli</td>
</tr>
</tbody>
</table>

Table 5.13: New classifier and lexical stem combinations: Combinations which occur at least twice (# ≥ 2).
SHOVE(29) and SHOVE:RR(30) really denotes *come/bring back to life*. What remains unclear so far is why the reflexive/reciprocal version SHOVE:RR(30) is used in the causative constructions in (5.36), i.e., in a transitive event in which Jesus is the agent and a man or a dead girl are the patients. Whether this is a mistake in the corpus (the forms of the classifier stems SHOVE(29) and SHOVE:RR(30) are very similar) or really is the case needs to be tested in future fieldwork.

(5.36)  a. **Murrinh-Patha:** Mu Jesus Wurdankabirl (wurdan-ngkabirl, 3GS.SHOVE:RR(30).NFUT-bring.to.life) Kardu Wakal Nugarn Nigunu Kardu Palngun Mekmunurr **English:** Jesus Brings a Man Back to Life

b. **Murrinh-Patha:** Jesus Wurdankabirl (wurdan-ngkabirl, 3GS.SHOVE:RR(30).NFUT-bring.to.life) Kardu Mardinhpuy i Mampatha Kardu Palngun Matharr Kumulung Kanthidha Wurrini **English:** Jesus Gives Life to a Dead Girl and Heals a Sick Woman

The lexical stem *(ngka)*rdurdu is also relatively often found in the corpus in combination with the classifier stem BASH:RR(15) (20 times). There is no separate lexical entry in the database for ngkardurdu, but it is in the database as a synonym for another word meaning ‘gather together’. So similarly to the ngkabirl case, the meaning is known but the database does not specify with which classifier stems it combines. The analysis of the corpus shows that ngkardurdu indeed means ‘gather’, as can be seen in (5.37), and that it combines with the classifier stem BASH:RR(15).

(5.37) **Murrinh-Patha:** Bere kardu wurnangart nimin-ya pubena-na-ngkardurdu-dha (3PLS.BASH:RR(15).PIMPF-3SGM.BEN-gather-PIMPF) ngarra Jesus i murrinh pubinanayewuptha pardi. **English:** Very many people were gathering around Jesus and listening to his words.

The lexical stem *(we)*wal shows the difficulties of an approach that solely works with this kind of corpus. Although the lexical stem occurs seven times in the corpus with the classifier stem *sit*(1), it is not really possible to determine from the corpus whether the lexical stem is wal or *we*wal, with a lexicalized incorporated body part *we* ‘head’. All the occurrences of *wal* also include *we*, so that it is very likely that the incorporated body part has lexicalized in this case. However, this conclusion is only tentative.

Moreover, the example shows that seven occurrences may not be enough to derive the meaning of the verb from the context. The verb occurs only
in contexts that are very loosely translated. (5.38) is the shortest context of the seven occurrences. In the longer contexts, it is even harder to determine which parts of the Murrinh-Patha sentences belong to which translation. However, even in this short context the meaning of the verb cannot be established. The Murrinh-Patha sentence contains two verbs, *dimpunpewal* and *banardurdidha*. The verb *dimpunpewal* consists of the third singular non-future form of the classifier stem *sit*(1) and the third person plural direct object marker *pun*, which is used for human direct objects only (Nordlinger 2011). The problem here is that the English translation does not contain a plural third person human direct object so that it is hard to determine what the verb actually means in this case.


**English**: But I tell you that even Solomon with his riches was not dressed as beautifully as one of these flowers.

That the pure number of occurrences does not necessarily matter for the interpretation is illustrated by the lexical stem *rerrarr*. The database lists *rerrarr* with the classifier stem *slash*(23) meaning ‘cut with knife’. In the corpus, *rerrarr* occurs four times with the classifier stem *break*(11) which is used for actions in which the hands are tearing something apart. All four occurrences are used to describe the well-known instance where Jesus tears apart bread with his hands, as can be seen in (5.39). The combination of the classifier stem *break*(11) with the lexical stem *rerrarr* is thus one of the fully compositional complex predicates which makes it easy to predict the meaning of the new combination.

(5.39) **Murrinh-Patha**: I bere nukum-ka mi *munganterrarr* (mungam-rerrarr, 3sgS.break(11).nfut-cut) i ku ngurlmirl-ka manyerryerr warda nukum-yu.

**English**: Then Jesus divided the food and gave it to his followers.

This concludes the discussion about the interpretation of the output of the second lookup strategy, i.e., the new combinations of classifier and lexical stems. In the following, I discuss the output of the morphological guesser, i.e., the new candidates for lexical stems.

### 5.4.2 New lexical stem candidates

The previous section provided a qualitative evaluation of the output of the second lookup strategy, i.e., of the unknown classifier and lexical stem combinations. It was shown that the output can be grouped into various subgroups
and that for many new combinations a meaning could be deduced from the context. This section now discusses the output of the morphological guesser. The guesser is used to detect new candidates for lexical stems. Similarly to the output of the second strategy, the output of the morphological guesser can also be sorted into different subgroups. The groups chosen for the output of the morphological guesser are spelling alternatives, lexical stems that resemble other known lexical items and lexical stems that do not belong to these two groups.

Table 5.14 lists examples of spelling alternatives found in the corpus. As can be seen from the table, the most common spelling alternatives involve alternations between voiced and voiceless sounds, e.g., yeekdhek vs. yegdhek, and alternations between retroflex and non-retroflex sounds, e.g., lalarr vs. rlarlarr.

In most cases, the alternatively spelled lexical stems occurred with the classifier stems that were listed in the database as well. For example, the lexical stem darrithuk occurred with the classifier stem POKE(19) in the corpus and the lexical stem rdarrithuk is listed in the database with the classifier stem POKE(19), too.

In some cases, however, the spelling alternative found in the corpus combines with a different classifier stem as is listed in the database. This is for example the case for the lexical stem wirnte which was found in the corpus with the classifier stem POKE(19) but wirnde is listed with the classifier stem POKE:RR(21) in the database. This is indicated in the Table by “+POKE:RR(21)”. As these classifier stems form a transitive and reflexive/reciprocal pair as discussed above, it is very likely that the lexical stems in question are indeed only spelling alternatives of each other.

What remains to be verified in future work, of course, is the question whether the spelling alternatives are indeed real alternatives or whether these are just mistakes in the corpus. An indication might be given by the number of occurrences. That is, if the number of occurrences is low, it is very likely that the spelling alternative found in the corpus is a mistake. This then offers the possibility to correct the mistakes in the corpus. However, if the spelling alternatives have a relatively high number of occurrences, such as the lexical stem mutmut, then this might be a true variety, in this case of the lexical stem mumut. Additional evidence for the assumption that both lexical stems mumut and mutmut may be in use with the same meaning comes from the fact that both can be considered reduplications of the lexical stem mut. The only difference then is that mutmut is a complete reduplication while mumut is a partial reduplication. Which of the alternatives is more commonly used must be established in future work.

In the second group of guessed lexical stems, the lexical stems bear a resemblance to other lexical items whose meaning is known. An overview over this group is provided in Table 5.15. These guessed lexical stems bear resemblances to adjectives or lexical stems in the database without prefixes that look like incorporated body parts.
<table>
<thead>
<tr>
<th>Lexical stem</th>
<th>CS</th>
<th>#</th>
<th>known LS (+CS)</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutmut</td>
<td>POKE(19)</td>
<td>13</td>
<td>mutut</td>
<td>dammutmut, dampirramutmut</td>
</tr>
<tr>
<td>yirrarart</td>
<td>POKE(19)</td>
<td>12</td>
<td>yirrat</td>
<td>dam-pirra-yirrarart</td>
</tr>
<tr>
<td>ngkarlbirl</td>
<td>SHOVE(29)</td>
<td>4</td>
<td>ngkarlbirl + SLASH(23)</td>
<td>ngurdun-karlbirl-nu, wurdan-pun-karlbirl</td>
</tr>
<tr>
<td>winharrarr</td>
<td>CS 17</td>
<td>4</td>
<td>winharrarr + BASH(14)</td>
<td>buniwinharrarrdha</td>
</tr>
<tr>
<td>winturt</td>
<td>POKE:RR(21)</td>
<td>11</td>
<td>winturt + POKE(19)</td>
<td>dempinturt, perremka-winturt-neme</td>
</tr>
<tr>
<td>winturt</td>
<td>POKE(19)</td>
<td>8</td>
<td>winturt</td>
<td>dempinturt</td>
</tr>
<tr>
<td>lalarr</td>
<td>HANDS(8)</td>
<td>3</td>
<td>rlarr</td>
<td>numa-lalarr-nenu</td>
</tr>
<tr>
<td>winhingkayit</td>
<td>POKE(19)</td>
<td>2</td>
<td>winhingkayith</td>
<td>dam-pun-pinhing-kayit</td>
</tr>
<tr>
<td>wirnte</td>
<td>POKE(19)</td>
<td>2</td>
<td>wirnde + POKE:RR(21)</td>
<td>daniwirtedha</td>
</tr>
<tr>
<td>yekdhek</td>
<td>BE(4)</td>
<td>2</td>
<td>yegdje</td>
<td>narnanyekdhek</td>
</tr>
<tr>
<td>darrithuk</td>
<td>POKE(19)</td>
<td>1</td>
<td>rdarrithuk</td>
<td>dam-nan-darrithuk</td>
</tr>
<tr>
<td>dhuurthurr</td>
<td>FEET(7)</td>
<td>1</td>
<td>thurthurr</td>
<td>punnudhurthurruunu</td>
</tr>
<tr>
<td>gurdugurdruk</td>
<td>BE(4)</td>
<td>1</td>
<td>gurdugurdruk</td>
<td>pardigurdugurdruk</td>
</tr>
<tr>
<td>ndarriwup</td>
<td>SIT(1)</td>
<td>1</td>
<td>rdarriwup + SLASH(24)</td>
<td>/ CS 25 dinindarriwuptha</td>
</tr>
<tr>
<td>ngkarlbirl</td>
<td>SHOVE:RR(30)</td>
<td>1</td>
<td>ngkarlbirl + SLASH(23)</td>
<td>puddingkarlbirlnu-yu</td>
</tr>
<tr>
<td>thap</td>
<td>HANDS:RR(10)</td>
<td>1</td>
<td>dhap + HANDS(8)</td>
<td>menthanmethap</td>
</tr>
<tr>
<td>thuth</td>
<td>SHOVE:RR(30)</td>
<td>1</td>
<td>thut</td>
<td>ngurdam-narra-thuth-ka</td>
</tr>
<tr>
<td>wirntigat</td>
<td>POKE(19)</td>
<td>1</td>
<td>wirntigat</td>
<td>parrampirntigat</td>
</tr>
<tr>
<td>wirntiway</td>
<td>POKE:RR(21)</td>
<td>1</td>
<td>wirntay</td>
<td>thempirntiway</td>
</tr>
<tr>
<td>yeypup</td>
<td>CS 16</td>
<td>1</td>
<td>yeypup</td>
<td>binaryeyupunu</td>
</tr>
</tbody>
</table>

Table 5.14: New lexical stem candidates: Spelling alternatives. The table lists the lexical stems as found in the corpus (first column) and provides the classifier stems (CS) with which the lexical stems were found in the corpus. It then lists the number of occurrences (#) and the lexical stem that is known from the database. If the lexical stem is not known with the classifier stem in question, the classifier stem for which the combination is known is added in this column as well.
Table 5.15: New lexical stem candidates: Lexical stems with similarity to other lexical stems (repeated forms, lexicon stems without incorporated body parts, etc.) or other lexical items.

<table>
<thead>
<tr>
<th>Known Lexical Item</th>
<th>Examples</th>
<th>Known Lexical Item</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ngurruthputh</td>
<td>hands(8)</td>
<td>numa-ngurruth-puth</td>
<td>known as LS namngurruthputh, numa-ngurruth-puth</td>
</tr>
<tr>
<td>rikerdek</td>
<td>poke:rr(21)</td>
<td>perrem-nu-marikerdek-kerdek-ka</td>
<td>known as LS perrem-nu-marikerdek-kerdek-ka</td>
</tr>
<tr>
<td>rdarrarr</td>
<td>hands:rr(10)</td>
<td>mangirdarrilenu, mardarrile-deyida-nu</td>
<td>known as LS mangirdarrilenu, mardarrile-deyida-nu</td>
</tr>
<tr>
<td>ngkat</td>
<td>lie(2)</td>
<td>slash(23)</td>
<td>known as LS nguwangkatnu, thuwangkat</td>
</tr>
<tr>
<td>rdarrath</td>
<td>ngka+rdarrath</td>
<td>mam-punku-rdarrath-nem</td>
<td>known as LS mam-punku-rdarrath-nem</td>
</tr>
<tr>
<td>thungku</td>
<td>poke:rr(21)</td>
<td>fire demthungku</td>
<td>is NC for fire demthungku</td>
</tr>
<tr>
<td>thutthut</td>
<td>thutthutthut</td>
<td>panthutthutnu</td>
<td>known as LS panthutthutnu</td>
</tr>
<tr>
<td>tumtum</td>
<td>poke:rr(21)</td>
<td>demtumtum</td>
<td>known as Adj demtumtum</td>
</tr>
<tr>
<td>werlerl</td>
<td>shove(29)</td>
<td>nuddurrawerl</td>
<td>known as LS nuddurrawerl</td>
</tr>
<tr>
<td>yetheth</td>
<td>hands(8)</td>
<td>snatch(9)</td>
<td>known as LS menanu-yetheth-tha</td>
</tr>
<tr>
<td>yitthathpirr</td>
<td>hands:rr(10)</td>
<td>nume-nung-kuyi-thathpirr-nem</td>
<td>known as Adj nume-nung-kuyi-thathpirr-nem</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Known Lexical Item</th>
<th>Examples</th>
<th>Known Lexical Item</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hands(10)</td>
<td>(10)</td>
<td>Hands(10)</td>
<td>(10)</td>
</tr>
<tr>
<td>Hands(8)’/Snatch(9)</td>
<td>(9)</td>
<td>Hands(8)’/Snatch(9)</td>
<td>(9)</td>
</tr>
<tr>
<td>Hands(8)’/Shovel(29)</td>
<td>(29)</td>
<td>Hands(8)’/Shovel(29)</td>
<td>(29)</td>
</tr>
<tr>
<td>Hands(8)’/Poke:rr(21)</td>
<td>(21)</td>
<td>Hands(8)’/Poke:rr(21)</td>
<td>(21)</td>
</tr>
<tr>
<td>Hands(8)’/Slash(23)</td>
<td>(23)</td>
<td>Hands(8)’/Slash(23)</td>
<td>(23)</td>
</tr>
<tr>
<td>Hands(8)’/Demtumtum</td>
<td>(29)</td>
<td>Hands(8)’/Demtumtum</td>
<td>(29)</td>
</tr>
<tr>
<td>Hands(8)’/Nuddurrawerl</td>
<td>(29)</td>
<td>Hands(8)’/Nuddurrawerl</td>
<td>(29)</td>
</tr>
<tr>
<td>Hands(8)’/Menanu-yetheth-tha</td>
<td>(29)</td>
<td>Hands(8)’/Menanu-yetheth-tha</td>
<td>(29)</td>
</tr>
<tr>
<td>Hands(8)’/Nume-nung-kuyi-thathpirr-nem</td>
<td>(29)</td>
<td>Hands(8)’/Nume-nung-kuyi-thathpirr-nem</td>
<td>(29)</td>
</tr>
</tbody>
</table>

Table 5.15: New lexical stem candidates: Lexical stems with similarity to other lexical stems (repeated forms, lexicon stems without incorporated body parts, etc.) or other lexical items.
An example for lexical stems which resemble adjectives is the guessed lexical stem `balbalk` which seems to combine with the classifier stem 17. In the database, `balbalbalkmam` is listed as an adjective meaning ‘flimsy, floppy’. Similarly, the adjective `kunungingki` ‘small, little’ is used as a lexical stem with the classifier stem `HANDS:RR(10)` in the corpus. In this case, the context can even verify that the meaning of the adjective is carried over to the meaning of the lexical stem. (5.40) provides the paragraph in which `kunungingki` is used as a lexical stem. More closely translated the second sentence is probably ‘and he raises up those who are small’.

(5.40) **Murrinh-Patha:** I kardu pana deyida wanku kardu pule ngala ngala-ya pepe da matha pantthuththutnu kardu nhini-yu. I kardu pana ngarra ngathayida **pumem-nu-kunungingki** (**3PLS.HANDS:RR(10)-RR-small**) pumpan-ka nukunu-ka man-ngala-nu warda nhini-yu.

**English:** God brings down rulers from their thrones, and he raises up the humble.

An example for a guessed lexical stem which is similar to other known lexical stems is `rdarrath`. The database lists the lexical stem `ngkardarrath` with the classifier stem `HANDS(8)`. `ngka` is an incorporated body part meaning ‘eye’ and, as was mentioned above, Murrinh-Patha lexical stems are often lexicalized combinations of incorporated body parts and other lexical stems. It seems that in this case, `rdarrath` might be the lexical stem which combines with the incorporated body part `ngka` to form the lexical stem `ngkardarrath` that is listed in the database.

What should be noted for all the lexical stems in Table 5.15, however, is the low number of occurrences in the corpus. That is, the likelihood that these are mistakes is relatively high. Nevertheless, as was seen in the example involving `kunungingki` above, some of these may indeed be examples of new lexical stems, or rather, that in certain contexts other parts of speech, such as adjectives, may be used as lexical stems. These examples then are a sign of the creative use of language and show that the class of lexical stems is not completely closed.

Finally, Table 5.16 lists all other guessed lexical stems which occur at least twice. A very frequent guessed lexical stem is *(ri)*numi, for which it is not clear whether the guessed lexical stem should be *numi* or *rinumi*. However, because the lexical stem always occurs with the incorporated body part *(ri)* ‘buttocks’, it is very likely that it is a lexicalized form and that the
Table 5.16: New lexical stem candidates. Lexical stem candidates which occur at least twice (# ≥ 2).

<table>
<thead>
<tr>
<th>Lexical Stem Candidates</th>
<th>Examples</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>kurdurinumi, nguddurinuminu</td>
<td>shove(29)</td>
<td>23</td>
</tr>
<tr>
<td>puddurinuminemenuyu</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>karnimninthar, parninthar, mardap, parninthar</td>
<td>be(4)</td>
<td>9</td>
</tr>
<tr>
<td>yibimkamnum-yu, yungkamnumtha</td>
<td>lie(2)</td>
<td>7</td>
</tr>
<tr>
<td>banalangtha</td>
<td>bash(14) or tarn(11)</td>
<td>18</td>
</tr>
<tr>
<td>banninthalang-yu</td>
<td>lang</td>
<td>17</td>
</tr>
<tr>
<td>bmlang</td>
<td>tum</td>
<td>26</td>
</tr>
<tr>
<td>pan-pun-pulu</td>
<td>slash(23)</td>
<td>2</td>
</tr>
<tr>
<td>bennygkarnimninthar</td>
<td>bash:rr(15)</td>
<td>3</td>
</tr>
<tr>
<td>bennygkamim</td>
<td>bash:rr(15)</td>
<td>1</td>
</tr>
<tr>
<td>bennygkarnim</td>
<td>see(13) or bash:rr(15)</td>
<td>1</td>
</tr>
<tr>
<td>bennynuy gkarnim</td>
<td>poke(19) or take(22)</td>
<td>1</td>
</tr>
<tr>
<td>kiwurdup, piwurdup</td>
<td>langma</td>
<td>25</td>
</tr>
<tr>
<td>piri-l-langma-langma</td>
<td>go(6)</td>
<td>1</td>
</tr>
<tr>
<td>wurrini-na-langma-dha</td>
<td>go(6) or cs(14)</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>go(6) or bash(14) or cs(13)</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>go(6) or bash(14) or cs(12)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>go(6) or bash(14) or cs(11)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>go(6) or bash(14) or cs(10)</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>go(6) or bash(14) or cs(9)</td>
<td>23</td>
</tr>
</tbody>
</table>
lexical stem is indeed rinumi. Although this lexical stem occurs 23 times in the corpus, it is quite difficult to guess its meaning from the context. This is so because the complex predicate formed with rinumi almost always occurs in conjunction with the complex predicate formed with the classifier stem slash(23) and the lexical stem bat which is translated as ‘hit/kill’ in the database. In the corpus, the conjunction is translated as ‘put to death/kill’, so that the meaning of the complex predicate formed with rinumi does not become clear.

(5.41) Murrinh-Patha: I kardu nangkal ngatha mere murrinh marru-bath-nu-ngintha ngarra yileyle i kalekale-ka i murrinh manku-rdarrithak-nu-ngintha-ka kardu nhini-ka nu-bat-bat-
(2PLS.slash(23).fut-kill-Fut) i nuuddu-rinumi-nu
(2PLS.shove(29).FUT-lexical.stem-FUT).

English: Anyone who says cruel things to his father or mother must be put to death.

The lexical stem lang is a nice example of what the corpus approach can actually achieve if the translations are not as free as in this Bible corpus. By coincidence, many translations of the instances involving lang are quite short and all of them involve instances of people spreading news and talking. An example is provided in (5.42). As a consequence, its meaning can be established quite surely as ‘spread news, talk’.

(5.42) Murrinh-Patha: Mu penintha nhini-ka da pana-re nanintha-lili-dha dini i murrinh nhini ban-nintha-lang-yu
(3sgS.17.NFUT-DU.M-spread.news-DM).

English: But the blind men left and spread the news about Jesus all around that area.

In conclusion, this section looked at the output of the second and forth lookup strategy in detail and tried to establish how much of the meaning of a new verb can actually be deduced from the context. The patterns that were established in Chapter 3 were helpful for this purpose. Nevertheless, quite often it was difficult to deduce the meaning and the exact syntactic behavior of the new verb from the context. The main reason for this lies in the fact that the corpus mostly provides very free translations. However, some instances of close translations also showed that it is in fact possible to deduce the meaning of the verb from the context if the translations are close enough. This is a helpful insight for the treatment of other languages or should other corpora of Murrinh-Patha become available. But even with this kind of freely translated corpora, the morphological analyzer can help identify new candidates for lexical items which can then be tested in fieldwork.
5.5 Conclusion

This chapter presented the challenges of a computational morphological analysis of Murrinh-Patha verbs and described a rule-based analyzer which was implemented using XFST and LEXC. With the help of these mechanisms, the complexity of the Murrinh-Patha verb could be implemented as closely as required.

The complexity of the Murrinh-Patha verb makes a corpus study that operates on the surface form of words very difficult. The different surface forms of the classifier stem and the high number of different morphemes that can be part of the Murrinh-Patha verb result in a very high number of different surface forms of verbs in any given text. To be able to analyze a text and abstract away from the different surface forms, for example to consider all verbs including the same classifier and lexical stem, a morphological analysis is needed. This can be achieved with the implementation presented in this chapter. The stepwise lookup strategy of the morphological analyzer can be used to ensure broad coverage and high precision at the same time. It is able to extract new candidates for classifier and lexical stem combinations and new candidates for lexical stems. In this way, it can be used to generate hypotheses for new verbs which can be tested in future fieldwork.

The morphological implementation can also be put to use in different applications. The morphological analyzer is used in an electronic dictionary that performs a morphological analysis for the user. The stepwise lookup strategy is advantageous in this case as the lookup strategies 2 and 4 ensure a more fine-grained user feedback in the electronic dictionary.

The morphological analyzer is also used in an XLE parser (Crouch et al. 2015) for Murrinh-Patha. The XLE parser is integrated into the ParGram community with standardized morpho-syntactic features and analyses (Butt et al. 1999). The XLE parser together with the morphological analyzer is then used in a small translation system from English into Murrinh-Patha. The translation system as well as the electronic dictionary is primarily intended as a learning resource for non-Murrinh-Patha speakers. A high degree of accuracy is therefore needed, which can only be accomplished with a rule-based system.

The implementation of the Murrinh-Patha XLE parser along with the different applications of the parser and the morphology are discussed in the following chapter in more detail.
Chapter 6

An xLE grammar for Murrinh-Patha and applications

The previous chapter described a computational morphological analyzer and how the analyzer can be put to use to process a Murrinh-Patha corpus. Because Murrinh-Patha is morphologically very complex, a corpus is very difficult to use without a morphological analyzer. It was shown that a careful modeling of the morphological facts can help analyze the corpus, both with respect to already known morphological facts as well as with respect to new insights.

For using the morphological analyzer, however, the user has to know how to handle the lookup utility of \textsc{xfst}. Similarly, the output of the morphological analyzer can only be interpreted by people who know how the morphological analyzer works and who have a good idea about the morphological complexities of Murrinh-Patha. It cannot be used by someone who is just beginning to learn Murrinh-Patha, for example.

This chapter presents a range of applications that help learners of Murrinh-Patha to grasp the complexity of the Murrinh-Patha system. These applications include an electronic dictionary that presents the user with a decomposition and the meaning of a word. A machine translation system is designed to translate from English into Murrinh-Patha, with special focus on the complex number marking found in Murrinh-Patha. Additionally, some smaller applications have been built in.

All of these applications have been built on top of a computational grammar of Murrinh-Patha. The computational grammar is implemented with the grammar development platform \textsc{xle} (Crouch et al. 2015) and is combined with the \textsc{xfst} morphology presented in the previous chapter.

Computational implementations of Australian languages are very rare. To my knowledge, computational implementations have only been designed
for the Australian languages Warlpiri, Wambaya and Arrernte. For Warlpiri, an early syntactic parser was presented by Brunson (1988), which made use of the morphological analyzer proposed by Sproat & Brunson (1987). A different syntactic parser for Warlpiri, in the framework of Government and Binding, was presented by Kashket (1986). For Wambaya, a grammar was implemented within the LinGO Grammar Matrix in the HPSG framework by Bender (2010). Finally, the implementation of Arrernte (Dras et al. 2012, Lareau et al. 2011) uses XLE and is thus most similar to the Murrinh-Patha implementation. However, while Murrinh-Patha is a non-Pama-Nyungan language, Arrernte is a Pama-Nyungan language and therefore differs in important aspects from Murrinh-Patha. In addition to serving as the basis for the applications, the implementation thus also functions as a test ground of how well a polysynthetic, morphologically complex language like Murrinh-Patha can be modeled computationally.

This chapter presents the XLE grammar of Murrinh-Patha and the applications that were built with it. It starts out with a brief description of XLE and the Murrinh-Patha grammar in Section 6.1. A complete description of the grammar would be beyond the scope of this thesis, but the basic outline of the grammar will be described, with special emphasis on the morphology-syntax interface assumed in the implementation.

Section 6.2 describes the various applications. It begins with a brief summary in Section 6.2.1 of why computational applications make sense for Murrinh-Patha. The section describes the language situation in the community in which Murrinh-Patha is spoken and briefly discusses the complexities from a language learning perspective. After this discussion, Section 6.2.2 describes the electronic dictionary while Section 6.2.3 is concerned with the translation system from English into Murrinh-Patha. Finally, Section 6.2.4 summarizes a range of smaller applications which almost automatically fall out from the implementation of the grammar. Section 6.2.5 concludes the chapter.

### 6.1 An XLE Grammar for Murrinh-Patha

The applications that are described in Section 6.2 all use XLE grammars of Murrinh-Patha and English to analyze the user input. Before these applications are described, this section is concerned with the implementation of the Murrinh-Patha grammar. It starts out in Section 6.1.1 with a brief overview over XLE grammars in general and describes the ParGram (Parallel Grammar) community, whose goal is to design XLE grammars of different languages with an agreed upon set of features and the same analyses for the same phenomena across languages.

After the outline of XLE grammars in general, the following subsections deal with the implementation of the Murrinh-Patha grammar in particu-
lar. Special emphasis is given to selected phenomena that arise because of Murrinh-Patha’s polysynthetic, head-marking characteristics.

The morphology-syntax interface that is assumed in the implementation is discussed in detail in Section 6.1.2. Many XLE grammars are combined with finite state morphologies to keep the lexicon simple (Kaplan et al. 2004). For a polysynthetic language like Murrinh-Patha, this is especially important. The section outlines Butt & Kaplan’s (2002) theoretical suggestion for the morphology-syntax interface and its adaptation to Murrinh-Patha. Section 6.1.3 then discusses how this general outline can be applied to implement Murrinh-Patha’s complex number system.

Section 6.1.4 addresses the implementation of Murrinh-Patha verbs, i.e., how combinations of classifier and lexical stems, but also classifier stems used as sole verbal predicate and used as auxiliary can be implemented. It is further concerned with the implementation of the complex tense and aspect system found in Murrinh-Patha.

Finally, Section 6.1.5 discusses the implementation of free word order in Murrinh-Patha and the challenges this poses for the interpretation of the dependents.

6.1.1 XLE grammars and the ParGram community

XLE is a grammar development platform on the basis of Lexical-Functional Grammar (Crouch et al. 2015, Butt et al. 1999). It has been used for the computational implementation of grammars of a whole range of different languages. Many of these implementations were carried out within the ParGram community, which strives to build parallel grammars in the implementations. As Butt et al. (1999:1) formulate it, “the grammars are parallel in the sense that they are guided by a common set of linguistic principles and a commonly agreed upon set of grammatical analyses and features”. As they explain, even within one linguistic theory, different analyses are often possible for a given phenomenon in a specific language. However, a cross-linguistic perspective might help to choose one analysis over another. This is not to say that every construction should receive the same analysis for all languages. There might be reasons to choose one analysis for one language and another analysis for another language. However, the ParGram community strives to give the same analyses to constructions which show the same properties in different languages. To ensure this, the community usually meets twice a year to compare and discuss their implementations. Sulger et al. (2013) present a parallel treebank for a set of sentences from ten different languages as output from these meetings.

As Butt et al. (1999) explain, the development of parallel LFG grammars with XLE started in 1994 with grammars for three languages: English (e.g. Riezler et al. 2002), German (e.g. Berman & Frank 1996, Dipper 2003) and French (e.g. Berman & Frank 1996). After this initial stage, grammars
An xle grammar for Murrinh-Patha and applications

for Japanese (e.g. Masuichi & Ohkuma 2003), Norwegian (e.g. Dyvik et al. 2005) and Urdu (e.g. Bögel et al. 2009) were added (Butt et al. 2002). Since then, bigger and smaller grammars have been developed for a whole range of typologically diverse languages, for which Müller (2013) provides an overview. To these languages belong, for example, Hungarian (Laczkó et al. 2010), Georgian (Meurer 2009), Indonesian (Arka et al. 2009), Irish (Sulger 2010), Malagasy (Dalrymple et al. 2006), Welsh (Mittendorf & Sadler 2005), Wolof (Dione 2012) or the Australian language Arrernte (Dras et al. 2012).

As XLE is a grammar development platform based on LFG, implementations usually consist of a lexicon and phrase structure rules in the formalism of LFG, as described in Chapter 1. The implementation is then able to provide LFG analyses of the input. For example, Figure 6.1 provides an example output of the English ParGram grammar for the sentence *He saw her*. The output consists of the f-structure as an attribute-value matrix (upper figure) and the c-structure represented as a tree (lower figure). This kind of output can now be used in a range of different applications, for example for corpus analysis or other applications which are described in the following section.

The XLE grammars can be used for generation as well. That is, the f-structure in Figure 6.1 can be used to generate the sentence *He saw her*, using the same or similar rules in the XLE implementation. Like parsing, generation can be very useful in various applications, e.g., it can be used to generate text from tables. Lareau et al. (2011) suggest this, for example, for English and Arrernte newspaper texts of Australian Football results.

The combination of parsing and generating allows the implementation of machine translation systems based on XLE (e.g. Frank 1999). The translation system for this purpose uses the underlying idea of the LFG formalism that the f-structure is a representation of deep syntactic analysis that is the same or similar across languages. When translating a sentence from English into Murrinh-Patha, for example, the English grammar is used to parse the sentence. This results in the c-structure and f-structure representation of the English sentence. Special rewrite rules can now be applied to the f-structure to transfer the English f-structure to an f-structure representing the Murrinh-Patha sentence. From this f-structure, the Murrinh-Patha sentence can be generated. Section 6.2.3 discusses the implementation of a translation system from English to Murrinh-Patha in more detail.

The special rewrite rules used to manipulate the f-structure in the translation process are part of a Packed Rewriting System and are often referred to as the Transfer System (Crouch et al. 2015). Besides machine translation, these transfer rules can also be used to rewrite f-structures into semantic representations, as was suggested by Crouch & King (2006) and applied for example by Zarrieß & Kuhn (2010), Hautli & Butt (2011) and Lareau & Dras (2012).

The XLE grammar development platform is thus very powerful, flexible and has been tested on a wide range of diverse languages. Nevertheless,
"He saw her."

```
PRED 'see[21:he], [104:sh]e''
  PRED 'he'
SUBJ NTYPE [NSYN pronoun]
  21 CASE nom, GEND-SEM male, HUMAN +, NUM sg, PERS 3, PRON-TYPE pers
  PRED 'he'
OBJ NTYPE [NSYN pronoun]
  104 CASE obl, GEND-SEM female, HUMAN +, NUM sg, PERS 3, PRON-TYPE pers
CHECK [SUBCAT-FRAME V-SUBJ-OBJ]
TNS-ASP MOOD indicative, PERF --, PROG --, TENSE past
CLAUSE-TYPE decl, PASSIVE --, VTYPE main
```

Figure 6.1: Example output of the English ParGram grammar for the sentence *He saw her*. On the top: f-structure; on the bottom: c-structure output.

Murrinh-Patha with its free word order and its complex morphological system poses some challenges. These challenges are discussed in the following subsections.

### 6.1.2 The general outline for the morphology-syntax interface

Murrinh-Patha as a highly polysynthetic and head marking language poses a range of challenges that have not been in the focus of ParGram so far.
The following sections describe these challenges and how XLE can be used to address them. This section is concerned with the general outline of the integration of the finite state morphological analyzer into XLE. The following sections then address specific challenges that the complex verbal template in Murrinh-Patha poses for the morphology-syntax interface.

The interface between morphology and syntax has been a matter of debate, both for theoretical linguistics and for grammar implementation (see, e.g., the discussions in Sadler & Spencer 2004). Polysynthetic languages pose special challenges for this interface because information otherwise associated with words, phrases and clauses is encoded in a single morphological word. Looking at the details of the computational implementation of a highly polysynthetic language such as Murrinh-Patha might thus help shed light on the problem.

XFS T morphologies are used in many XLE grammar implementations (Kaplan et al. 2004). As Frank & Zaenen (2004) point out, such morphologies make a lexicon with fully inflected forms unnecessary. This is crucial for a polysynthetic language, as listing all inflected forms for a language like Murrinh-Patha would be infeasible, if not impossible. Kaplan et al. (2004) describe the general process of how an XFS T morphology can be incorporated into an XLE grammar. While it is thus clear how the general mechanism works, some challenges concerning the division of labour between the XFS T morphology and the XLE grammar in details of the implementation remain.

This section first presents the theoretical assumptions for the morphology-syntax interface as laid out by Butt & Kaplan (2002). It then shows how this architecture is put to use in the computational implementation of the Murrinh-Patha XLE grammar, following the approach described by Kaplan et al. (2004) for XLE grammars in general.

Butt and Kaplan’s (2002) theoretical proposal for the interface

Butt & Kaplan (2002) propose a realizational model of morphology that passes on morphological information to the syntax.1 For this purpose, they define a complex relation $R$ to model the interface between morphology and syntax. More precisely, $R$ is a relation that “realizes the morphological features of a given f-structure as a string: $fRw$” (Butt & Kaplan 2002:3). They propose to decompose the complex relation $R$ into two relations called $Sat$ and $D$:

$$R = Sat \circ D.$$ 

The relation $Sat$ is the satisfaction relation holding between an f-structure and an f-description as already proposed by Kaplan & Bresnan (1982). The description relation $D$ models the relation between the f-description and

---

1 An earlier version of this section was published in Seiss (2011).
Butt & Kaplan (2002) display $D$ as a set of ordered pairs $\langle f\text{-description, sequence} \rangle$ and use (6.2) as an example for the string /walks/ which is associated with f-descriptions concerning the subcategorization frame of the verb as well as number, person and tense information.

$$(6.2) \quad \langle \{ (f_1 \text{ pred}) = \text{walk}, (f_1 \text{ subj}) >, (f_1 \text{ subj pers}) = 3, (f_1 \text{ subj num}) = s_g, (f_1 \text{ tns-asp tense}) = \text{pres} \}, /\text{walks}/ \rangle$$

As Butt & Kaplan (2002) point out, the relation called $D$ is traditionally considered the morphology-syntax interface and how this interface should be modeled formally has been the subject of much debate. They propose to model the interface by decomposing $D$ further into a lexical relation $L$ and a sequence relation $Seq$, which renders (6.3) as the overall decomposition of the complex relation $R$.

$$(6.3) \quad R = \text{Sat} \circ L \circ Seq$$

The lexical relation $L$ maps between f-descriptions and what Butt & Kaplan (2002) call description-names (D-names). Examples are given in (6.4). These D-names are atomic symbols (with arbitrary names, but for convenience mnemonic terms are chosen) and are, in the sequence relation $Seq$, linked to the string, e.g. as in (6.5).

$$(6.4) \quad \begin{align*}
\text{walk:} & (\uparrow \text{ pred}) = \text{walk} <(\uparrow \text{ subj}) > \\
3: & (\uparrow \text{ subj pers}) = 3 \\
\text{Sg:} & (\uparrow \text{ subj num}) = s_g \\
\text{Pres:} & (\uparrow \text{ tns-asp tense}) = \text{pres}
\end{align*}$$

$$(6.5) \quad \langle \{ 3, \text{sg, walk, pres} \}, /\text{walks}/ \rangle$$

The described architecture thus associates a string (or phonological word) with an f-structure via a set of D-names and f-descriptions. The use of D-names makes this approach a realizational model, in which the relationship between affixes and their functions can be quite complex.

To summarize, the sequence relation $Seq$ maps a set of D-names to a string, determining which strings are possible in a given language. It can thus be considered the morphological part of the relation. The satisfaction relation $Sat$ maps between an f-structure and an f-description and is thus part of syntax, while the lexical relation $L$ is the mapping between the morphological D-names and the syntactic f-descriptions and can thus be considered the morphology-syntax interface.

The morphology-syntax interface in the Murrinh-Patha grammar

The theoretical architecture as laid out in Butt & Kaplan (2002) was transferred to the computational implementation of the integration of finite state
morphologies into XLE grammars by Kaplan et al. (2004). In the following, the integration as proposed by Kaplan et al. (2004) is discussed for the integration of the Murrinh-Patha XFST morphology into the XLE grammar.

The mechanisms that XFST offers have been already described in Chapter 5 and will not be repeated here. However, what is important for the XLE implementation is the output of the morphology in which a string is associated with a number of tags to encode information. In (6.6), the surface form *bamkardu* is associated with the information that this form is made up of the stem *bam*, which is classifier stem number 13 in its third person singular non-future form, an unexpressed third person direct object, and the lexical stem *ngkardu*, marked by +LS. The relation between the string and the tags (D-names) is thus the instantiation of the sequence relation \( \text{Seq} \).

\[(6.6) \quad \text{bamkardu} : \]
\[\text{bam} + \text{class13} + 3P + \text{sgS} + \text{nFut} + 3\text{sgDO} + \text{ngkardu} + \text{LS}\]

The morphology output serves as input for XLE and thus needs to be interpretable by the syntax. For this purpose, the D-names need to be associated with f-descriptions. An example of an excerpt of a ‘morphological lexicon’ instantiating the lexical relation \( L \) is given in (6.7). These are referred to as ‘sublexical entries’ in the XLE system. The D-name +class13, for example, contributes the information that the classifier stem is number 13 and the tag +3P that the subject is third person. In a similar way, stems such as *bam* and *ngkardu* receive sublexical entries as well, which will be discussed more detailedly in Section 6.1.4. The second column specifies the category, similar to lexical entries in XLE and LFG.

\[(6.7) \quad +\text{class13 } \text{CLASS} \quad (\uparrow \text{cs}) = 13.\]
\[+3P \quad \text{PERS} \quad (\uparrow \text{subj pers}) = 3.\]

The XLE system also needs to know how to combine these sublexical entries, which is modeled in sublexical rules. A simplified sublexical rule for verbs is given in (6.8). In this rule, CS\(_{BASE}\)\(^2\) marks the place for a classifier stem and CLASS\(_{BASE}\) is used for the tags +classX. The rule further specifies that verbs consist of a tag for person information (PERS\(_{BASE}\)) as well as number and tense information and further tags that are contributed by the verbal template slots 2 to 4. LexS\(_{BASE}\) provides the place for the lexical stem, +ngkardu for example, and LexStem\(_{Base}\) for the tag +LS. The dots represent further tags that can be contributed by the verbal template slots 6 to 9.

\[(6.8) \quad V \rightarrow \text{CS\(_{BASE}\) CLASS\(_{BASE}\) PERS\(_{BASE}\)} \ldots\]
\[\quad \text{LexS\(_{BASE}\) LexStem\(_{BASE}\)} \ldots\]

\(^2\)\(_{BASE}\) is added in the sublexical rules to the categories given to the tags and stems in the sublexical entries as a naming convention within XLE.
An xle grammar for Murrinh-Patha and applications

205

Morphology

Word formation:
- concatenation of morphemes
- constraints of verbal template
- morpho-phonological processes

<table>
<thead>
<tr>
<th>Morphology-Syntax Interface</th>
<th>Argument structure of classifier and lexical stem combinations</th>
<th>Complex number system interpretation</th>
<th>Complex tense interpretation</th>
<th>Sublexical rules and entries in XLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>Word order</td>
<td>Phrase structure rules in XLE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: Overall architecture of the computational approach

In this way, the sublexical rules mirror the output of the XFST morphology. It is important to note that the constraints the verbal template poses are modeled in the XFST morphology and that it is only the syntactic interpretation that is modeled in the syntax-morphology interface of the XLE system.

An overview over the architecture of the computational approach is presented in Table 6.1. The implementation of the morphology with XFST and LEXC was the topic of Chapter 5. The basics of the implementation of the morphology-syntax interface were laid out in this section. In the following, I describe how complex morphology-syntax interfaces can be modeled with sublexical rules and entries within XLE. Finally, section 6.1.5 addresses the issue of how head-marking and free word order can be implemented with phrase structure rules in the Murrinh-Patha XLE grammar.

6.1.3 The morphology-syntax interface for the complex Murrinh-Patha number system

The use of D-names modeling the morphology-syntax interface allows for simple mappings such as described in (6.7) in the previous section but also for more complex mappings between the syntax and the morphology. Such a complex mapping is needed for the Murrinh-Patha complex number system, for which neither a morpheme-based account nor a simple mapping between morphology and syntax is possible. This is discussed in the following.

The complex Murrinh-Patha number system was introduced in Chapter 1 and 5. Here, only the facts which provide evidence for a complex
morphology-syntax interface are briefly reviewed. As discussed by Nordlinger (2010a, 2012b), subject number is determined by a combination of morphemes, i.e., the classifier stem and special number markers, e.g., for the dual female non-sibling subject, the special number marker is \(-\text{ngintha}\). The relevant facts are illustrated in (6.9). If the singular classifier stem appears on its own, the subject is indeed singular. However, if the singular classifier stem is combined with \(-\text{ngintha}\), the subject is specified as being dual female non-sibling.

(6.9) a. \textit{bampardu}  
\texttt{bamp-ngkardu}  
\texttt{3SGS.SEE(13).NFUT-see}  
‘He/she saw him/her.’ (Nordlinger 2012b)

b \textit{bamp-\text{ngintha}-ngkardu}  
\texttt{3SGS.SEE(13).NFUT-DU.F-see}  
‘They two (fem non-sibl.) saw him/her.’ (Nordlinger 2012b)

Nordlinger (2010a) initially proposed to model these facts with a morpheme-based analysis which is given in (6.10). In this analysis, singular classifier stems either determine that the subject number is singular or require (=c) the information that the subject number is dual to be provided from a different source. This information can be given by the number marker \(-\text{ngintha}\), for example.

(6.10) \texttt{bamp:}  
\texttt{↑ subj num} = sg v (↑ subj num) = c du  
\texttt{-\text{ngintha}:}  
(↑ subj num) = du  
(↑ subj gend) = fem  
(↑ subj sib) = –

With this approach, Nordlinger (2010a) also captures agreement facts between the complex predicate and serialized posture and motion classifier stems. The serialized posture and motion classifier stems have to agree with the classifier stem in tense as well as subject person and number, not with the complete verb. This means that they agree with the morphological form of the classifier stem, and actually not with the syntactic realization of the number encoded in the whole verb. This can be seen in (6.11) in which the complex predicate involves the dual marker \(-\text{ninth}\) but the serialized classifier stem \texttt{sit(1)} does not involve a dual marker.

(6.11) \texttt{manganintharta}\texttt{=}\texttt{dim}  
\texttt{mangan-ninth-rta}\texttt{=}\texttt{dim}  
\texttt{3SGS.SNATCH(9).NFUT-DU.M-hug=}\texttt{3SGS.SIT(1).NFUT}  
‘They two (boys, non-siblings) are hugging her.’ (Nordlinger 2010a)
For such examples, the second alternative, namely that the information that the subject number is dual needs to be provided from a different source, applies for the serialized posture or motion verb as well and the dual number marker provides the number specification for the whole combination.

However, as Nordlinger (2012b) lays out, a morpheme-based analysis as proposed in (6.10) is difficult as the dual number markers ngintha and nintha can also be used as object markers as in (6.12). In this case, the dual object marker -nanku together with -ngintha determines that the object is dual female non-sibling.

\[(6.12)\text{ ma-nanku-rdarri-purl-nu-ngintha} \]
\[1\text{SGS.HANDS}(8).\text{FUT-2DU.DO-back-wash-FUT-DU.F} \]
\[\text{‘I will wash your (dual non-sibling) backs.’} \quad (\text{Nordlinger 2012b})\]

Nordlinger (2012b) further points out that the use of -ngintha as subject or object number marker can lead to ambiguities in certain cases. In fact, (6.12) is ambiguous between the readings specified in (6.13). In (i), -ngintha is interpreted as object number marker while in (ii) it is interpreted as subject number marker. The interpretation in (iii) can evolve as the subject and object number markers share the same slot in the verbal template. Thus, if -ngintha is interpreted as object number marker in (iii), no slot is left in the verbal template in which subject number could be encoded (c.f. also Nordlinger 2010c).

\[(6.13)\text{ ma-nanku-rdarri-purl-nu-ngintha} \]
\[1\text{SGS.HANDS}(8).\text{FUT-2DU.DO-back-wash-FUT-DU.F} \]
\[(i) \text{‘I will wash your (dual non-sibling) backs.’} \]
\[(ii) \text{‘We (du.exc.nsib) will wash your (du.sib) backs.’} \]
\[(iii) \text{‘We (du.exc.nsib) will wash your (du.nsib) backs.’} \]
\[\text{ (Nordlinger 2012b)}\]

As discussed by Nordlinger (2012b), this data is problematic both for a morpheme-based analysis as proposed by Nordlinger (2010a) as well as for constructive number approaches such as Arka (2011). Nordlinger (2012b) therefore instead argues for an approach in which the context of the whole morphological word is taken into account. This can be achieved by combining the output of the XFST morphology with a complex morphology-syntactic interface.

As has been discussed in Chapter 5, a distinction can be made between -ngintha/-nintha that is interpreted as subject number information versus object number information with the help of flag diacritics in the XFST morphology. For this purpose, the XFST morphology uses word-internal constraints such as the number information of the classifier stems or the dual object marker -nanku. This means that the disambiguation of subject and
An xle grammar for Murrinh-Patha and applications

Object number marking already completely happens in the morphology and that the xle system only receives ambiguous input from the morphology if the verb is indeed ambiguous such as e.g., (6.13).

If the verb is not ambiguous, xle receives unambiguous input. For the sentence in (6.11), for example, the morphology passes on to the syntax that *dim* involves a singular subject while *mangan-nintha-rta* involves a dual male non-sibling subject, i.e., it produces the output in (6.14).³

(6.14)  
a. dim:  
dim +class13 +3P +sgS +nFut  
b. mangan-nintha-rta:  
mangan +class9 +3P +sgS +nFut +du.m.non-Sibl.S  
+3sgDO +rta +LS

To avoid a clash of the differing number information in syntax, a complex morphology-syntax interface is needed that combines Nordlinger’s (2010a) complex interpretation of morphemes with the realizational approach as presented by the output of the xfst morphology. That means that the information that was associated with the morphemes in Nordlinger’s (2010a) approach is now associated with the D-names for the number information, as can be seen in (6.15).

(6.15)  
+sgS:  
(↑ subj num) = sg v  
(↑ subj num) =c du & (↑ subj sibl) =c –  
+du.m.non-Sibl.S:  
(↑ subj num) = du  
(↑ subj gend) = masc  
(↑ subj sib) = –

The D-name *+du.m.non-Sibl.S* has a simple f-description while the D-name *+sg* receives a complex f-description that either specifies that its number is singular (if no other subject number marker is present) or that it requires a dual non-Sibling subject (as in (6.11)). The constraint (↑ subj sibl) =c – has been added to Nordlinger’s (2010a) analysis to block the combination of singular serialized posture and motion verbs combining with complex predicates formed with dual sibling classifier stems.

To sum up, a realizational morphology in combination with a complex morphology-syntax interface is needed to model the complex agreement facts of the Murrinh-Patha number system.

³The information that it is the singular classifier stem form in (6.14) (marked by +sgS) is retained to be able to use the morphological analyzer as a stand-alone application. As will be seen below, it does not interfere with the syntactic analysis.
6.1.4 Implementing Murrinh-Patha verbs

A further question for the implementation of Murrinh-Patha is how verbs should be implemented, i.e., how their occurrence and their subcategorization frames should be modeled. The implementation for this purpose has to address three points: it has to specify which classifier stems can function as the sole verbal predicate and how their subcategorization frames should look like; it has to model which classifier stems can function as serialized posture and motion verbs and has to determine how these should be implemented; finally, the implementation has to address the question how classifier and lexical stems should be implemented.

As has been laid out in the previous section already, the xfst morphology provides the input for the XLE grammar in that it associates words with tags. These tags are then interpreted in the XLE grammar in sublexical entries and sublexical rules. The sublexical rule for Murrinh-Patha complex predicates has already been provided in (6.8). To model the fact that Murrinh-Patha verbs can also consist of certain classifier stems without a lexical stem, an alternative structure is added to the rule in (6.16).

\[(6.16) \ V \rightarrow \ {\text{CS\_BASE} \ \text{CLASS\_BASE} \ \text{PERS\_BASE} \ldots \ \text{LexS\_BASE} \ \text{LexStem\_BASE} \ldots \ | \ \text{V\_BASE} \ \text{CLASS\_BASE} \ \text{PERS\_BASE} \ldots } \]

The first two lines of the rule model the structure of classifier lexical stem combinations while the third line models the structure of Murrinh-Patha verbs in which the classifier stems function as the sole verbal predicate. In addition, sublexical entries for the classifier and lexical stems are needed to determine the subcategorization frames for the verbs.

For the classifier stems which can be used as the sole verbal predicate, the tag for the classifier stem is associated with a pred value and a subcategorization frame in the sublexical entry, just like normally verbs would be associated with in a lexical entry. For example, the classifier stem sit(1) in its third person singular form dim specifies that it is an intransitive verb if used on its own, as can be seen from the sublexical entry in (6.17) which makes use of the template as specified in (6.18).

\[(6.17) \ \text{dim} \ V \ @(\text{Intrans} \ \text{dim} \ 1). \]

\[(6.18) \ \text{Intrans}(P \ N) = \{(\uparrow \text{PRED}) = 'P<(\uparrow \text{SUBJ})>' \]
\[\ (\uparrow \text{SUBJ PRED})=’PRO’ \ |
\ (\uparrow \text{PRED}) = P<(\uparrow \text{SUBJ})>' \}
\[\ (\uparrow \text{VTYPE MAIN}) = + \]
\[\ (\uparrow \text{VTYPE CLASSIFIER-STEM}) = N. \]

In the template, the variable P is replaced by dim as the reference name for the classifier stem sit(1) and N is replaced by 1 to additionally encode the
classifier stem number. The template further models an optional PRO for the subject with the standard LFG analysis for pro drop.

The implementation of the classifier and lexical stem combination is not as straightforward, as these involve complex predicates. It is thus not directly obvious how the classifier and lexical stems should be combined and how the subcategorization frames for the combinations should be modeled.

As was discussed in the previous chapters, complex predicate formation in other languages is usually seen as an operation of the argument structure. Such complex predicates are then usually implemented in XLE with a restriction operator (Butt et al. 2003) that allows the manipulation of argument structure in the process of combining the different parts of the complex predicate. The restriction operator is usually called up by the syntactic or sublexical rule that combines the light verb with the other elements forming the complex predicate. It specifies how the argument structure of the complex predicate should be formed based on the light verb and the argument structure of the other element. The restriction operator is thus very useful for the implementation of productive complex predicates, i.e., in which light verbs combine with a large group of similar lexical items, e.g., with all transitive verbs etc.

As has been discussed in the previous chapters, Murrinh-Patha complex predicates are restricted in their combinatory possibilities through semantic concepts, not through their argument structure. This actually means that Murrinh-Patha classifier and lexical stems cannot combine as freely as complex predicates in other languages. Every classifier stem only has a restricted range of lexical stems that it can combine with. In this case it does not make sense to use the restriction operator for the implementation.

The theoretical analysis presented in Chapter 4 proposed to analyze Murrinh-Patha complex predicates with the help of LCS blueprints and hierarchies of selectional restrictions. The hierarchies of selectional restrictions presented a realizational approach to the semantics of the combination in that they made explicit which meaning aspects were responsible for the restriction on the combination, e.g., the manner of motion. The LCS blueprints functioned as the level of lexical semantics that is linked to the argument structure and represented a primitive-based approach in that they made explicit which meaning parts were contributed by the lexical stem and the classifier stem respectively.

As the XLE grammar concentrates on the implementation of the syntax with the aim of developing practical applications for learners of Murrinh-Patha and as the XLE system does not offer a straightforward way to implement the lexical semantic level, LCS blueprints and hierarchies of selectional restrictions are not part of the XLE implementation. However, the implementation nevertheless captures the same underlying idea, namely that the classifier and lexical stem together determine how they combine, and that the regularities of the combination are captured in templates. In the im-
plementation, this means that various templates are used which model how the subcategorization frames of the classifier and lexical stem combinations should look like.

For this purpose, the tags for the lexical stems receive sublexical entries as specified in (6.19a). The sublexical entry for the lexical stem specifies what kind of complex predicate it forms in combination with a specific classifier stem. For example, the tag \( +ngkardu \) for the lexical stem \( ngkardu \) in (6.19a) specifies that it forms a transitive complex predicate (CP\_DO) with the classifier stem \( \text{see(13)} \).

\[
\begin{align*}
(6.19) \quad a. \quad +ngkardu & \quad \text{LS @}(\text{CP\_DO ngkardu 13}). \\
\quad b. \quad \text{CP\_DO (LS CS)} & = \quad (\uparrow \text{pred}) = \text{LS} < (\uparrow \text{subj}) (\uparrow \text{obj})> \\
& \quad (\uparrow \text{cs}) = \text{c CS}.
\end{align*}
\]

The template for the transitive complex predicate is specified in (6.19b). It has the variables LS and CS which are filled by the lexical stem and the classifier stem as specified in the sublexical entry for the lexical stem in (6.19a). The template specifies that the lexical stem requires a subject and an object argument. That this subcategorization frame is only used if \( ngkardu \) combines with the classifier stem \( \text{see(13)} \), for example, is ensured by the constraining equation \((\uparrow \text{cs}) = \text{c CS}\) which requires that the information \((\uparrow \text{cs}) = 13\) is contributed from another sublexical entry. This information is then contributed by the tag for the classifier stem. This means that in the implementation, the classifier and lexical stems are listed in the lexicon as combinations and determine the subcategorization frame together, but the regularities of their combinations are modeled in the templates. This reflects the theoretical analysis in which each classifier and lexical stem combination specifies which LCS blueprint it uses.

In the Murrinh-Patha implementation, different templates are defined to model intransitive and transitive complex predicates as well as complex predicates requiring an indirect object. Of course, as the implementation does not use LCSs to model argument structure, fewer templates are needed in the implementation in comparison to the LCS blueprints proposed in Chapter 4.

In addition to the classifier and lexical stem combinations discussed in this thesis, the computational implementation also includes further types of verbal predicates. The XLE grammar is able to parse instances of impersonal verbs (Walsh 1987, Nordlinger 2010a) which are combinations of...
"bamkardu"

| PRED         | 'ngkardu<[-1-SUBJ:PRO], [-1-OBJ:PRO]' |
| SUBJ         | [PRED 'PRO']                     |
| NUM sg, PERS 3 |
| OBJ          | [PRED 'PRO']                     |
| NUM sg, PERS 3 |
| CHECK        | [DO pres, _TELICITY telic, _TENSE non-fut] |
| TAM          | [ASPECT perf, MOOD indicative, _TENSE past] |
| VTYPE        | [CLASSIFIER-STEM 13, COMPLEX-PRED +] |
| -1 CLAUSE-TYPE decl |

Figure 6.2: F-structure as generated by the XLE grammar for bamkardu

classifier and lexical stems in which the logical subject is denoted by object morphology. An implementation of so-called voums and nerbs (Walsh 1996), morphologically intermediate categories between nouns and verbs used as predicates syntactically, also belongs to the implementation of verbal predicates in the Murrinh-Patha grammar.

Finally, the implementation also models which classifier stems can be used as serialized posture and motion verbs. As Nordlinger & Caudal (2012) point out, classifier stems 1-7 can encliticize onto the complex predicate to change the tense, aspect and mood interpretation of the verb, as can be seen in (6.20), in which the classifier stem sit(1) serializes onto a complex predicate in non-future tense. Without the serialized classifier stem, a past perfective interpretation arises ((6.20a), while the complex predicate receives a present imperfective reading if it is combined with the serialized classifier stem ((6.20b)).

(6.20) a. dirraninthanubath
   dirran-nintha-nu-bath
   3SGS.WATCH(28).NFUT-DU.M-RR-look.at
   'They (two) looked at each other.'

b. dirraninthanubath=dim
   dirran-nintha-nu-bath=dim
   3SGS.WATCH(28).NFUT-DU.M-RR-look.at=3SGS.SIT(1).NFUT
   'They (two) are looking at each other.'

(Nordlinger & Caudal 2012)

These serialized classifier stems have been implemented as auxiliaries in
the XLE system. This means that a classifier stem such as sit(1) actually has three different interpretations: it can either function as the sole verbal predicate, be part of a complex predicate or it can be used as an auxiliary. This is specified in the sublexical entry in (6.21).

\[(6.21)\] \[\text{dim} \ V \ \text{@(Intrans dim 1):} \]
\[\text{CS} \ (↑\text{vtype classifier-stem}) = 1; \]
\[\text{AUX} \ (↑\text{check SERIAL CLITIC}) = +.\]

The auxiliary has been implemented as forming a VP constituent with the complex predicate. This rule is, however, quite complex, as the interpretation of the tense, aspect and mood category also depends on the telicity of the complex predicate, as Nordlinger & Caudal (2012) discuss. That is, the same morphological marking results in different tense realizations, as can be seen in the contrast in (6.20a) vs. (6.22). In (6.22), the non-future classifier stem is interpreted as present imperfective while it is interpreted as past perfective in (6.20a). As has been discussed above already, in order for (6.20a) to receive a present imperfective reading, an intransitive classifier stem has to be added as a clitic, such as dim in (6.20b).

\[(6.22)\] wurrang-nintha-lili
\[\text{3SGS.GO(6).NFUT-DU.M-walk} \]
‘They are walking.’

Nordlinger & Caudal (2012) model the different syntactic interpretation of the same morphological marking by assuming that the telicity of the complex predicate plays a role for the interpretation. Atelic complex predicates such as (6.22) receive a present imperfective interpretation while telic complex predicates such as (6.20a) receive a past perfective interpretation if the classifier stem is inflected for non-future tense. A different behavior of telic and atelic complex predicates can also be observed for complex predicates with past imperfective marking on the classifier stem.

What this means is that every sublexical entry for a classifier and lexical stem combination has to specify whether the combination is telic or atelic. To establish the telicity of the classifier and lexical stem combinations, the examples provided in Street (1989), along with their translations, have been used as a database. For example, if the translation of an example involving a non-future classifier stem was present imperfective, the complex predicate was specified as being atelic. This was carried out for all combinations of classifier and lexical stems, i.e., for transitive and intransitive complex predicates as well as for impersonal verbs. Table 6.2 provides an overview over the numbers of atelic and telic complex predicates implemented in the grammar.

The implementation of the telicity of the complex predicates again makes use of templates. A complex predicate such as in (6.22) has the sublexical
entry in (6.23a). This sublexical entry makes reference to the telicity and the subcategorization frame of the complex predicate. It calls the template @LS\textsubscript{Intrans\_Atelic} which is provided in (6.23b). This template specifies that the telicity is atelic (which is used to model the right tense and aspect information in combination with the morphological tense information provided by the classifier stem) and calls the template for intransitive complex predicates. This template is provided in (6.23c).

(6.23) a. +lili \textsc{LS} @(LS\textsubscript{Intrans\_Atelic} lili lili 6).

b. LS\textsubscript{Intrans\_Atelic}(LS CS) = @(LS\textsubscript{Intrans} LS CS)

\((↑ \text{CHECK \_TELICITY}=\text{atelic}.)\)

c. LS\textsubscript{Intrans}(LS CS) = (↑ \text{PRED}) = 'LS <(↑ \text{SUBJ})> '
\((↑ \text{CS}) = _\text{c} \text{CS}.)\)

That templates can make reference to other templates has been proposed by Dalrymple et al. (2004) for an even more efficient implementation. How the templates make reference to each other can be displayed in template hierarchies. For the Murrinh-Patha verbs, different telic and atelic template versions are needed for all different subcategorization frames. This template hierarchy is displayed in Figure 6.3. With the help of these templates, the classifier and lexical stem combinations can be implemented efficiently. Additionally, the template hierarchy shows what factors play a role in the syntax of classifier and lexical stem combinations.

The telicity of the complex predicate is specified as a special \textsc{check} feature in the \textsc{xle} implementation. As Crouch et al. (2015) specify, “the \textsc{check} feature is a feature that each grammar can use for grammar internal features that are largely used as well-formedness checks” and can thus be considered the equivalent to the \textsc{m}-features proposed by Sadler & Spencer (2001). Similarly, the tense specifications of the classifier stems and the presence of the serialized posture or motion verbs are also implemented as \textsc{check} features as they only serve to define the tense realizations in combination with the telicity of the complex predicate.

<table>
<thead>
<tr>
<th>Complex predicate</th>
<th>Telic</th>
<th>Atelic</th>
</tr>
</thead>
<tbody>
<tr>
<td>intransitive</td>
<td>219</td>
<td>472</td>
</tr>
<tr>
<td>transitive</td>
<td>942</td>
<td>85</td>
</tr>
<tr>
<td>with indirect object</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>impersonal verb</td>
<td>75</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 6.2: Classifier and lexical stem combinations implemented in the Murrinh-Patha \textsc{xle} grammar.
The interplay of all these factors in then modeled in the VP rule. A fragment for the VP rule for the classifier stem in non-Future tense is given in (6.24). The first choice specifies that the predicate is interpreted as past perfective if no auxiliary is present, i.e., if (↑ CHECK _SERIAL_CLITIC) = +, and if the complex predicate is specified as being telic. The second choice accounts for atelic complex predicates without an auxiliary. In this case a present imperfective reading arises. The third choice finally accounts for telic complex predicates which require an auxiliary to receive a present imperfective reading. In this way, the complex Murrinh-Patha tense system in which the morphological marking and the semantic interpretation do not have a one-to-one correspondence can be implemented in the xle system.

\[(6.24)\quad \text{VP} \rightarrow V: \uparrow=\downarrow\]
\[
\{(\uparrow \text{CHECK } _{\text{TENSE}}) = c \text{ non-fut} \}
\[
\{(\uparrow \text{TAM TENSE}) = \text{past} (\uparrow \text{TAM ASPECT}) = \text{perf} \quad (\uparrow \text{CHECK } _{\text{SERIAL_CLITIC}}) = + \quad (\uparrow \text{CHECK } _{\text{TELICITY}}) = c \text{ telic} \}
\[
(\uparrow \text{TAM TENSE}) = \text{pres} (\uparrow \text{TAM ASPECT}) = \text{imperf} \quad (\uparrow \text{CHECK } _{\text{SERIAL_CLITIC}}) = + \quad (\uparrow \text{CHECK } _{\text{TELICITY}}) = c \text{ atelic} \}
\[
(\uparrow \text{AUX}: \uparrow=\downarrow). \]

To sum up, the implementation of Murrinh-Patha verbs comprises classifier stems used as sole verbal predicates, classifier and lexical stem combinations as complex predicates and serialized posture and motion classifier stems which are implemented as auxiliaries.

The implementation of classifier and lexical stem combinations as complex predicates reflects the main ideas of the theoretical analysis presented
in Chapter 4: the classifier and lexical stems are listed in the lexicon as combinations and jointly specify the subcategorization frames of the complex predicates. Instead of LCS blueprints, templates are used to model the regularities found in the combinations.

The complex tense and aspect interpretation of Murrinh-Patha verbs made it necessary to specify an inherent telicity for each classifier and lexical stem combination. The VP rule then models the appropriate tense and aspect interpretation based on the morphological tense marking on the classifier stem, the inherent telicity of the classifier and lexical stem combination and the presence of the serialized posture or motion classifier stem. The Murrinh-Patha tense system is thus another example of the complexities of periphrastic verbal constructions such as e.g., discussed in Butt et al. (2004), Frank & Zaenen (2004), Ackermann & Webelhuth (1998), Börjars et al. (1997) and Sadler & Spencer (2001).

6.1.5 Implementing free word order in Murrinh-Patha and the interpretation of the dependents

Because Murrinh-Patha is a highly polysynthetic language, many issues that arise for the computational implementation of Murrinh-Patha address questions of the implementation of the morphology and the morphology-syntax interface, as was discussed above. For the Murrinh-Patha syntax as such, only simple sentences have been implemented in the XLE grammar. This means that the Murrinh-Patha grammar is quite small, especially compared to other languages, if phrase structure rules are considered. The grammar only has 16 rules. This has two main reasons. Firstly, while the complex morphology is well described for Murrinh-Patha, many questions remain open on the syntactic level and its interactions with other levels, such as e.g., with information structure. More theoretical research is needed for a broader implementation of the Murrinh-Patha syntax. Secondly, the implementation in XLE was motivated by the various applications for learners of Murrinh-Patha which will be presented in Section 6.2. As these applications were designed for beginning learners of Murrinh-Patha, emphasis was laid on the implementation of simple sentences in combination with a well-developed lexicon.

This section now addresses the implementation of the free word order that is found in Murrinh-Patha and the challenges this poses for the interpretation of the dependents. The rule in (6.25) is used to implement simple Murrinh-Patha sentences. It specifies that a Murrinh-Patha sentence has to consist of a VP, which was given in (6.24) and any number of other constituents, marked as X*. X may for example be a variable for an NP argument, but may also correspond to an NP or PP adjunct. The linear order of the constituents is free, which is modeled with the shuffle operator “,” (Crouch et al. 2015).
An xle grammar for Murrinh-Patha and applications

(6.25) \( S \rightarrow X^*, \text{VP} \)

Such a flat c-structure is usually assumed if the word order is free, e.g., by Austin & Bresnan (1996) or Nordlinger & Bresnan (1996). While the phrase structure as such is thus straightforward, the simplicity also bears challenges. The free word order in combination with optional case marking on the dependents and the different possibilities of building noun phrases results in many different analyses for one sentence. This is discussed in the following in more detail.

In Murrinh-Patha, case marking on the noun phrase is optional (Blythe 2009a). It is usually the inflection on the verb that indicates what the subject and the object is. However, if the subject and the object are both third person singular, for example, different possibilities for the interpretation of the noun phrases arise. This can be seen in (6.26). Because no case marking is present on the dependents, the noun phrase \( \text{ku baybaye} \) can either be the subject, as in (i), or the object, as in (ii).

(6.26) \[
\begin{align*}
\text{ku} & \quad \text{baybaye} & \quad \text{manganta} \\
\text{ku} & \quad \text{baybaye} & \quad \text{mangan-rta} \\
\text{NC}_{\text{anim}} & \quad \text{kangaroo} & \quad \text{3SG}.\text{SNATCH}(9).\text{NPUT-hug} \\
(i) & \quad \text{‘The kangaroo is hugging him/her.’} \\
(ii) & \quad \text{‘He/She is hugging the kangaroo.’} \\
(iii) & \quad \text{‘The kangaroo is hugging the thing from the ku class.’} \\
(iv) & \quad \text{‘The thing from the ku class is hugging the kangaroo.’}
\end{align*}
\]

The picture is further complicated by the fact that noun classifiers such as \( \text{ku} \) can function as nouns by themselves and nouns do not need noun classifiers to form noun phrases (Blythe 2009a). This may lead to further ambiguity as \( \text{baybaye} \) might be interpreted as the subject and \( \text{ku} \) as the object as in (iii). Similarly, \( \text{ku} \) can be interpreted as the subject and \( \text{baybaye} \) as the object as in (iv). This renders four different interpretations for the quite simple sentence in (6.26).

There is, however, an interpretational preference to interpret \( \text{ku baybaye} \) as one noun phrase, as “most noun phrases do have a classifier” (Blythe 2009a:103). XLE offers the possibility to model such preferences with optimality marks inspired by Optimality Theory (Crouch et al. 2015, Frank et al. 1998). The underlying idea is that a certain phrase structure rule or a lexical entry is associated with a certain optimality mark and the optimality marks are ordered based on their importance. When parsing an ambiguous input string, the output is ordered based on the ordering of the optimality marks of the phrase structure rules and lexical entries that were needed to construct the analysis.

For the Murrinh-Patha grammar, optimality marks have been used in the implementation for noun phrases that give priority to analyses in which
ku baybaye is interpreted as one noun phrase over the other analyses. This gives priority to (6.26i) and (6.26ii) over (6.26iii) and (6.26iv).

However, the ambiguity between the interpretation of ku baybaye as subject or object (i.e., between (6.26i) and (6.26ii)) is difficult to resolve in the same manner. Whether overtly expressed noun phrases are more often interpreted as subject or object in Murrinh-Patha still needs to be determined. If a clear tendency exists, this tendency could be implemented in the same manner using optimality marks. However, it is more likely that no clear tendency exists as the interpretation of the overt noun phrase depends on the discourse context. This then cannot be modeled within XLE, as knowledge about information structure, anaphora resolution etc. would be needed. These notions play a role in research on natural language processing that makes use of XLE, for example on generation tasks such as described in Zarrieß et al. (2012). However, in this line of research, XLE is usually used to produce a range of alternatives for which a stochastic ranker selects the most appropriate analysis. Such a procedure is, e.g., described by Zarrieß et al. (2012). Thus, the problem of ranking is left to a separate module and not incorporated into XLE in these approaches.

Such an additional ranker for Murrinh-Patha, which takes sentence-internal as well as sentence-external factors into account, is, however, beyond the scope of this work. This means that so far, the implementation treats both interpretations (6.26i) and (6.26ii) as equally likely and the user has to choose the one that is appropriate in the context.

6.1.6 Conclusion

This section discussed the implementation of the Murrinh-Patha XLE grammar. The section first presented an overview over XLE grammars and the ParGram community in general. It then moved on to discuss the syntax-morphology interface assumed in the XLE grammars and showed that while the general architecture is well established, the details of the implementation have to be adjusted to the phenomena under consideration. The complex Murrinh-Patha number system as well as the complex Murrinh-Patha tense system in which no one-to-one correspondence between the morphology and syntax and its semantic interpretation exists is a challenge for the implementation which could be mastered by using complex mappings between the different levels.

The implementation of the XLE grammar for Murrinh-Patha showed that it is possible to implement a highly polysynthetic, head-marking language such as Murrinh-Patha in XLE. The findings of this section and the previous chapter suggest that the modular architecture of LFG is well suited for a language like Murrinh-Patha in which the morphology and syntax follow very different rules. The Murrinh-Patha morphology displays a very complex verbal template in which strict interdependencies between the single
morphemes exist. In contrast, the Murrinh-Patha syntax is characterized by a high degree of freedom in its word order, case marking and discourse marking. LFG’s modular architecture allows for modeling these different properties with different sets of tools. This is one of the strengths of the LFG theory.

Similarly, the theory is also open towards a complex morphology-syntax interface, which can be modeled using Butt & Kaplan’s (2002) architecture. Because of Murrinh-Patha’s highly polysynthetic nature, the focus of the computational implementation naturally lies on the morphology and the morphology-syntax interface. Thus, in contrast to other grammars, the XLE Murrinh-Patha grammar is quite small when the phrase structure rules are considered. However, with these rules many simple sentences can actually be analyzed so that the Murrinh-Patha grammar can be used in a range of different applications. These applications are discussed in the following section.

6.2 Applications for learners of Murrinh-Patha

In this section, a range of applications for learners of Murrinh-Patha are described.4 These applications use the XLE grammar to analyze the user input and provide the user with appropriate output. Before these applications are presented, however, Section 6.2.1 briefly discusses why it makes sense to build such computational applications for Murrinh-Patha. It argues that both the Murrinh-Patha social situation as well as its language structure make designing applications interesting and valuable. The remainder of the section then presents an electronic dictionary for Murrinh-Patha (Section 6.2.2), an English to Murrinh-Patha translation system (Section 6.2.3) as well as further applications (Section 6.2.4).

6.2.1 Motivation

Computational applications are usually built for languages with a large amount of speakers for which it is certain that the application will be widely used. From this perspective, it does not seem to make sense to build computational applications designed to learn Murrinh-Patha, as probably not many people will want to learn Murrinh-Patha. However, application development as seen from a research perspective may also be beneficial as a test ground, e.g., to see how morphological complexities can be implemented in a morphological analyzer and in a computational grammar, and additionally how such implementations can be integrated into Computer-Assisted Language Learning (CALL) programs in general. Of course, it is also desirable that applications that are built for such purposes are then used in real life. I

---

4 An earlier version of this section was published as Seiss & Nordlinger (2012).
will argue in this section that Murrinh-Patha is actually one of the few Australian languages for which it makes sense to build such applications, both from the situation of the Murrinh-Patha speakers and from the complexity of the system.

Murrinh-Patha, as has been already stated in Chapter 1, is spoken by approximately 2500 people in and around Wadeye (Port Keats), a small community approximately 400 kilometers south of Darwin (cf. e.g. Nordlinger 2011). As has been documented by Kelly et al. (2010), the language is, despite its actual small number of speakers, not considered endangered by the Murrinh-Patha speakers themselves. This seems to be a reasonable self-assessment as the numbers of Murrinh-Patha speakers compiled by the Australian Bureau of Statistics census have been rising from 1430 speakers in 1996 to 1832 speakers in 2006 (Kelly et al. 2010). Additionally, Taylor (2010) estimates that the number of people living in Wadeye (and therefore most probably speaking Murrinh-Patha) in the year 2029 will figure between 3500 and 4000. Murrinh-Patha is thus one of the few Australian languages that is not considered endangered (Nordlinger 2011).

The reasons why Murrinh-Patha is not endangered at the moment are discussed by Kelly et al. (2010). They point out that Murrinh-Patha is the first language of most of the speakers in the community. While older community members (45+) have reasonable knowledge of English as a second language due to their mission experiences, younger speakers have only limited knowledge of English. Consequently, the language of daily interaction in the community is Murrinh-Patha and parents expect that children will learn English at school. Wadeye is relatively isolated, which means that there is little incidental traffic or visitors and consequently, English is not often used in daily life.

However, some English speaking people such as school teachers or nurses usually live in Wadeye for a shorter or longer period of time. The electronic applications which are presented in this chapter are intended for people from this group who would like to learn some initial vocabulary of Murrinh-Patha beyond the simplest phrases. In a further step, however, the implementation of the Murrinh-Patha grammar could also be used to design CALL applications for Murrinh-Patha speakers to learn English.

From this perspective then it makes sense to build computational applications for Murrinh-Patha. It should be pointed out that of course Murrinh-Patha is not the only Australian language for which such applications make sense. For the Australian language Arrernte, for example, a system is currently being designed that can generate texts describing Australian Football League (AFL) games (Lareau et al. 2011).

The complex verbal structure of Murrinh-Patha is another reason why applications for beginning Murrinh-Patha learners make sense. A significant problem that a beginning language learner has to face is that the complex verbal structure makes using a paper dictionary for Murrinh-Patha very
difficult. As was seen throughout the thesis, the meaning of the verb is determined jointly by the classifier and the lexical stem. Street (1989), and also Reid & McTaggart (2008) for Ngan’gityemerri, list the lexical stems alphabetically in the dictionary and provide the meaning of the lexical stem depending on the classifier stem it combines with in the lexical entries.

However, users have to be very advanced in their understanding of the verbal structure to be able to use such a dictionary. They have to know how to decompose a verb into its various morphemes to extract the lexical stem and look it up in the dictionary. Due to the complex verbal morphology with the nine different template slots, extracting the lexical stem is very complicated in Murrinh-Patha. As has been discussed already, the lexical stem is deeply embedded inside the verbal structure as it occupies slot 5. What comes before the fifth slot is very variable as the 38 classifier stems have approximately 50 different surface forms each and the second, third and fourth verbal template slots can be filled by many different morphemes as well. Additionally, the surface form of a verb may not be the pure concatenation of morphemes as phonological rules may apply. These properties make extracting the correct lexical stem to look it up in a paper dictionary very difficult.

Alternatively, a paper dictionary could of course list all forms of the classifier stem with the lexical stem. However, this is impractical as this would involve over 50 entries (for all the forms of the classifier stem) of the same verb. Moreover, this does not solve the problem of the other morphemes which might intervene between the classifier and the lexical stem in the second, third and forth verbal template slot.

This shows that both options for a paper dictionary are problematic for a beginning learner of Murrinh-Patha. The electronic dictionary that is proposed in section 6.2.2 solves this problem as it performs a morphological decomposition for the user and provides the user with the lexical entry of the lexical stem in dependence of the classifier stem. The electronic dictionary thus covers the Murrinh-Patha to English direction.

For the English to Murrinh-Patha direction, the learner has to face even more complex challenges. The verbal template with nine different morphemes in itself is already quite complex and thus is difficult to learn. However, the system is even more complex because of the dependencies that exist between the different morphemes. A range of these dependencies have been discussed already, but one example is repeated in (6.27). The dual female subject number marker -nginthra usually attaches between the classifier and lexical stem as in (6.27a). However, if a direct object marker is present, e.g. -nhi in (6.27b), the subject marker can only be realized after the lexical stem. Such complex dependencies have to be understood in order to form a grammatically correct Murrinh-Patha verb.
(6.27) a. *bam-ngintha-ngkardu*
   1SG.S.see(13).NFUT-DU.F-see
   ‘We two saw it.’

b. *bam-nhi-ngkardu-ngintha*
   1SG.S.see(13).NFUT-2SG.DO-see-DU.F
   ‘We two saw you.’

(6.27) also illustrates that number marking is quite complex in Murrinh-Patha, as discussed above. The number system is encoded with a combination of different morphemes for the whole paradigm of subject, direct and indirect object marking. In addition to the complex encoding, the number system in itself is quite complex with a distinction between singular, dual, paucal (small groups) and plural, a distinction between sibling and non-sibling in the dual and paucal categories as well as a distinction between female and male in the non-sibling categories. Thus, the Murrinh-Patha number system in itself and its encoding is complex as there is no one to one correspondence between morphemes and the syntactic information they encode. This is difficult for a beginning learner of Murrinh-Patha.

Similarly to the encoding of different number information via a combination of morphemes, tense information can be encoded with a combination of different morphemes as well. As was shown by Nordlinger & Caudal (2012) and discussed above, the same morphological marking results in different tense information depending on the telicity of the classifier and lexical stem combination.

A learner of Murrinh-Patha thus does not only have to know the different surface forms for a classifier stem depending on the tense information, he or she also has to know whether the complex predicate is considered to be telic or atelic in order to form a grammatically correct verb or to encode the tense information that he or she wants to. Here again, there is no one to one correspondence between the morphemes and the syntactic information which makes an already complicated system even more complex.

The translation system that is proposed in Section 6.2.3 as well as the further applications presented in Section 6.2.4 target these points. They were especially designed to practice the complex number system by making explicit what factors play a role in the morphological composition. It also incorporates the various dependencies between the subject and object number encodings. Additionally, the implementation also adheres to the tense system as described in Nordlinger & Caudal (2012).

### 6.2.2 The Murrinh-Patha to English dictionary

The goals of the Murrinh-Patha to English electronic dictionary have been discussed in the previous section: the electronic dictionary should be able to perform the required morphological (and syntactic) analysis for the user
and should present the user with the relevant dictionary entries. This was accomplished by combining the xfst morphological analyzer and the XLE grammar for Murrinh-Patha with a user interface that is implemented with Perl TK. To use the dictionary and the other applications, users do not have to be acquainted with XFST or XLE.

The electronic dictionary can be used to look up Murrinh-Patha words, phrases and sentences. When the user types in a sequence of Murrinh-Patha words, this input is sent to the XLE grammar for analysis. From the XLE analysis, a script extracts the relevant information concerning the classifier and lexical stem as well as nouns, noun classes, adjectives etc. For these pieces of information, the relevant dictionary entries are extracted and presented to the user. The electronic dictionary uses Street (1989) as a database, so the information that the user receives is the meaning that is specified in Street (1989) for the lexical entries as well as examples that are also part of Street (1989).

Figure 6.4 presents an example output of the electronic dictionary. The user input in this case was kardu kigay bamkardu ‘the boy saw it’. This was analyzed by the XLE grammar and the relevant information is displayed as output. The user is presented with the analysis for the verb, namely that it is the lexical stem ngkardu ‘see’ in combination with the classifier stem 13, and the information for the noun phrase, namely that the noun kigay means ‘teenage boy’ and belongs to the noun class for Aboriginal people, kardu (Walsh 1997).

The electronic dictionary uses the XLE grammar as a lookup system instead of just the XFST morphology because it takes syntactic dependencies into account to restrict the possible interpretations. For example, the constructed example in (6.28) is very similar to the example above as it consists of a noun classifier, a noun and a verb as well. However, the noun classifier mi cannot form a noun phrase with the noun kigay as kigay belongs to the noun class kardu. The noun classifier mi therefore forms the object just as a full noun and the dictionary output therefore also provides a separate output for the noun classifier mi.

(6.28) mi kigay bamkardu
     mi kigay bam-ngkardu
     NC_reg boy 3SGS.SEE(13).NFuture-see
      ‘The boy saw it (fruit or vegetable).’

One challenge for electronic resources is to make the system robust, i.e., to provide useful output even if the system cannot analyze the user input completely. The electronic dictionary for this purpose makes use of the stepwise morphological analyzer that was presented in the previous chapter. The XLE grammar first uses the first lookup strategy, i.e., in which all known combinations of classifier and lexical stems are implemented. For
Type in a Murrinh-Patha word or phrase:

```
kardu kigay bamkardu
```

Search

Result for 'kardu kigay bamkardu':

Verb: 'ngkardu' + Classifier Stem 13:
'to see, to look'

Examples:

bangkardu mani 'I'll have a look'
pipi nginarr nhinhi damkardu
thurran? 'do you look at your wife's mother?'

Noun: 'kigay' + Noun Classifier 'kardu':
'teenage boy'

Figure 6.4: Output of the Murrinh-Patha electronic dictionary
these combinations, the output can be returned as shown in Figure 6.4 and discussed above.

However, if the user input cannot be analyzed as a known combination of classifier and lexical stem, the electronic dictionary uses the other lookup strategies to give the user more refined feedback about the reasons of the failure. Thus, if the user input can be decomposed in a classifier stem and a lexical stem whose combination is not in the database, it tells the user that the input can be decomposed in a classifier and lexical stem combination but that the meaning of this combination is unknown.

Similarly, the morphological guesser can be used if the user input cannot be decomposed in a combination of a classifier stem and a known lexical stem. In this case then, the guesser is used to guess the lexical stem as was described in the previous chapter. In such a case the user receives the output that the input could not be decomposed in a known combination of classifier and lexical stem and that the lexical stem had to be guessed.

These refined feedback possibilities could probably also profit from a matching of the user input with all possible strings that the morphological analyzer can generate. This would help most with typical typing errors or with spelling variations. Such a matching algorithm could be implemented using Levenshtein distance but is left for future work.

6.2.3 The English to Murrinh-Patha translation system

While the electronic dictionary covered the direction from Murrinh-Patha to English, the translation system targets the direction from English to Murrinh-Patha. It can translate simple English sentences into Murrinh-Patha and is especially intended as a resource which helps to find the correct verb form and to study the structure of the verbal complex. The translation system uses the standardly assumed architecture for xle translation systems as described in Section 6.1.1, but is combined with a disambiguation module that prompts the user to specify the number of people if an English plural form is used.

The architecture of the system is presented in Figure 6.5. The user input is analyzed by the English grammar, which builds an abstract representation of the sentence as its f-structure. The disambiguation module then checks whether a plural is present in the f-structure of the English sentence. If indeed a plural is present, the disambiguation module prompts the user to specify how many people are involved in the action, whether they are siblings and whether they are a group of purely male individuals, a group of purely female individuals or a mixed group. After the user has specified this information, the English f-structure is transferred into a corresponding Murrinh-Patha f-structure with the help of rewrite rules. Finally, the Murrinh-Patha grammar generates a valid sentence from the f-structure. This ensures that the Murrinh-Patha output is always grammatical, which is important in a learning system.
An example of the translation system is provided in Figure 6.6. The user input was “They saw me.” in this case. As the subject of the English sentence is plural, the disambiguation module asks the user whether the subject is a group of two, a small group or a bigger group. Because the user specified that the subject is a group of two, the disambiguation module further asks whether they are siblings. If the user had ticked “a big group”, the disambiguation module would not have asked for siblinghood, as this is irrelevant for big groups in Murrinh-Patha. Thus, the disambiguation module only asks for further specifications if they are required. In this way, the user can learn which factors play a role in which combination.

The system offers this disambiguation for subjects and objects and, of course, also if the subject and the object are both plural. It further prompts the user to specify whether the first person plural is inclusive or exclusive.
of the hearer. It so covers all possible number combinations and helps the learner to understand the differences of the various verb forms. In addition to the complex number system, the tense system which depends on the telicity of the classifier plus lexical stem combination is inbuilt into the translation system as well.
6.2.4 Further applications

The electronic dictionary and the translation system are the main applications that were designed for learners of Murrinh-Patha. However, the Murrinh-Patha implementation with XFST and XLE offers a range of more applications that basically fall out from the implementation for free. These further applications are described briefly here.

Firstly, the modular implementation of the XFST morphology allows an application that is dubbed “Phonological Form Checker”. It targets the problem that learners might know what morphemes they have to use to form a certain word, but that they still have problems with the phonological rules that apply. Some phonological rules are quite simple in Murrinh-Patha, but some are more complex and depend on the lexems used. Thus, if learners are not sure what the surface form of the word looks like, they can type in the concatenation of morphemes and the phonological form checker returns the surface form.

An example is provided in Figure 6.7. It illustrates the output for the standard example used in this work, the combination of the classifier stem see(13) with the lexical stem ngkardu ‘see’. The phonological rule that applies here specifies that /ng/ is lost if it follows /m/ or /n/. Thus, the output for the input string bam-ngkardu is bamkardu.

The phonological form checker makes use of two different morphological analyzers. For the morphological analyzer that is used to analyze the user input, the phonological rules are optional. That means that the analyzer can provide analyses for both forms, the concatenation of morphemes and the actual surface form. From the analysis that is provided by the first analyzer, the second analyzer generates the surface form. For the second analyzer, the phonological rules have been implemented as obligatory, so the second analyzer only generates the surface form and not the pure concatenation of morphemes. Because both morphological analyzers otherwise use the same implementation, the application actually did not need much extra work.

The other applications are offered to the user as more information when they use the translation system or the electronic dictionary. Figure 6.8 displays the possibilities when the learner uses the translation system and requests more information for the translation output bamningkardunintha. If the user wants it to, the system displays the dictionary entries and examples, he or she can look at the morphological information and part of the paradigm of the translation output can be displayed with all possible tenses, all possible subject numbers and all possible object numbers.

The electronic dictionary is offered here to the user as a more detailed description of the lexical entry and the examples might help the user to establish whether the translation given is in fact what was intended by the user. This option is only offered as more information if the learner used the translation system. If the learner used the electronic dictionary, he or she
is offered the possibility of generating the verb form with different tense, person and number information. This then uses the disambiguation system that was designed for the translation system. In this way, the same resources are again put to different uses for the learner.

The other further applications are the same for the translation system and the electronic dictionary. The display of the morphological information uses the xfst morphology of Murrinh-Patha for which the output has been rewritten in order to make it more understandable for a beginning learner of Murrinh-Patha.

The display of excerpts of the verbal paradigm in which the same verb can be shown with all possible tenses, with all possible subject numbers or with all possible object numbers can be used to get a more detailed overview over the verbal paradigm. Because Murrinh-Patha has so many different forms for the subject and object numbers, the paradigms that can be sensibly displayed for the user to understand and follow has to keep either the subject or the object number information constant. As Murrinh-Patha has seven different number categories both for the subject and the object, 49 different verb forms would be displayed otherwise.

These further applications thus offer the user various ways of investigat-
An xle grammar for Murrinh-Patha and applications

6.2.5 Conclusion

This chapter presented the Murrinh-Patha xle grammar and a range of applications that were designed using this grammar. The applications do not require knowledge of how xle or xfst is used. As such, they can be widely used by people who want to learn Murrinh-Patha. At the same
time, the applications might spark interest in the field of computational linguistics by demonstrating what computational linguistics can actually do with respect to underresourced and morphologically complex languages.

The applications presented in this chapter were intended for beginning learners of Murrinh-Patha, i.e., they do not require much prior knowledge of the complicated Murrinh-Patha verbal system. The main applications were the electronic dictionary, which uses the XLE grammar of Murrinh-Patha, and the translation system, which uses XLE grammars of Murrinh-Patha and English as well as rewrite rules for transfer. In both cases, the system uses deep language processing to perform the required tasks. This is important as such learning resources require a high degree of precision, i.e., the output of the applications should be grammatically correct so that people can actually use the application to study and learn the language. While the design of such deep language processing resources might be time consuming, they are nevertheless worthwhile as they guarantee high precision in their output. Additionally, the same resources can be used in different applications, such as the XLE grammar which is used in the electronic dictionary and the translation system. This also makes the development of these resources more efficient. Other applications could be built with the Murrinh-Patha XLE grammar as well, for example CALL applications that teach Murrinh-Patha speakers English.

So far, the user interfaces are implemented with Perl TK and the system requires a local installation. In order to make the applications more widely accessible, a web interface is planned for the different applications. This web interface could then also include an interface in which users could add new lexical items without knowledge of XFST and XLE.
An xle grammar for Murrinh-Patha and applications
Chapter 7

Conclusion

By looking at the complex nature of Murrinh-Patha verbs, this thesis discussed how computational linguistics and theoretical insights can be used to complement each other. Both my computational implementation and the theoretical analysis I proposed for Murrinh-Patha complex predicates profited highly from already existing theoretical descriptions of Murrinh-Patha such as Street (1989), Nordlinger (2010c) and Blythe (2009a). On the other hand, the theoretical analysis as well as the computational implementation generated new insights and regularities that can now be tested against even more data.

The computational implementation of the morphology, presented in Chapter 5, demonstrated what even a small corpus study can achieve. Because of the complex verbal morphology, such a corpus study is impossible without a morphological analyzer that abstracts away from the surface form of words. For the development of such a morphological analyzer, the existing resources, mainly Street (1989) and Nordlinger (2010c) have been used. The already known facts were implemented with the finite state technology tools xfst and lexc. The implementation showed that these tools are capable of handling such morphological complexities, as flag diacritics are a powerful extension of the finite state mechanism.

The analysis of the output of the morphological analyzer in the corpus study allowed for considerable additions to the morphological analyzer. New morpheme orderings could be detected which can be tested in future fieldwork. It was also possible to detect new combinations of classifier and lexical stems and even new lexical stems with the help of the morphological analyzer. These new verb candidates can be tested in future fieldwork as well. However, the detailed evaluation of the corpus showed that at least some candidates could be verified based on the corpus study alone.

The xfst morphology was also incorporated into an xle grammar of Murrinh-Patha, which was discussed in Chapter 6. The implementation showed that xle is well equipped for such challenges like the complex num-
ber and tense systems found in Murrinh-Patha. The chapter also laid out how the XLE grammar can be used in various applications such as in an electronic dictionary and a translation system.

The implementations with the rule-based systems both for the morphology and the syntax would not have been possible without a detailed understanding of the theoretical patterns involved. While the implementation of the morphology could build on various descriptions of the verbal template and other parts of speech such as Nordlinger (2010c) or Blythe (2009a), the various combination patterns of classifier and lexical stem combinations, necessary for an effective implementation of verbs with templates in XLE, had not been described before extensively and in detail.

Chapter 3 therefore was a necessary prerequisite for the implementation. It provided an overview over the various patterns found in Murrinh-Patha complex predicate formation from an argument structure perspective. It was argued that argument structure does not play a prominent role in the selection process of the combinatory possibilities of Murrinh-Patha complex predicates. That is, the argument structure of the classifier stem does not determine the argument structure of the complex predicate directly. For example, transitive classifier stems can be used in intransitive complex predicates and vice versa. It is also not the case that classifier stems just specify that the argument structure is determined by the lexical stem, which would be expected, for example, in aspectual complex predicates. Murrinh-Patha complex predicate formation is thus unrestricted from an argument structure point of view.

However, complex predicate formation is not unrestricted in Murrinh-Patha. Semantic concepts such as the shape of an instrument involved in the action restrict the combinatory possibilities. The selectional restrictions are thus more fine-grained than argument structure. This was discussed in Chapter 4 for a case study involving the classifier stems BASH(14), POKE(19) and SLASH(23) which showed that each of these classifier stems has several meaning dimensions that account for the selectional restrictions. The different combinatory possibilities were displayed in hierarchies of selectional restrictions. In contrast to previously described complex predicates, Murrinh-Patha complex predicates can be considered semantic complex predicates. Murrinh-Patha complex predicates thus need a different formal modeling for their argument structure.

It was argued that the groups that arise from the semantic restrictions can be used to derive the argument structure. This is accomplished via a formal modeling of the argument structure of Murrinh-Patha complex predicates by means of LCS blueprints. The blueprints model the idea that it is not the classifier stem alone that determines the basic shape of the complex predicate, as is assumed in most other approaches to complex predicate formation. Instead, the classifier stem and the lexical stem together determine the lexical conceptual structure of the complex predicate.
The use of blueprints further enables to assume underspecified sublexical entries for the classifier and lexical stems. In the approach presented in this thesis, classifier and lexical stems do not have a whole LCS decomposition which specifies the event and argument structure. In contrast, they may just specify certain semantic concepts such as the shape of an instrument for classifier stems or a certain result state for lexical stems. In this way, it is not necessary to assume that Murrinh-Patha classifier and lexical stems have a certain argument structure, which was shown to be difficult in Chapter 3. This also accounts for the fact that most classifier stems and all lexical stems cannot occur as sole verbal predicates in Murrinh-Patha.

As each classifier and lexical stem combination has to specify which LCS blueprint it uses, it is also not surprising that for certain combinations it is not clear why this specific classifier stem is used. Besides accounting for the productive combinations in an elegant manner, the LCS blueprint approach thus also explains the tight link between classifier and lexical stems in certain complex predicates in Murrinh-Patha and the tendency for lexicalized, opaque combinations.

Finally, it was argued that the use of LCSs for the argument structure of Murrinh-Patha provides the right granularity of information as they incorporate the most important concepts that are relevant in the selectional restrictions of Murrinh-Patha complex predicates. In contrast to thematic role information, for example, LCSs assume that more semantic information is relevant for the mapping to syntax. This is needed for Murrinh-Patha complex predicate formation, as semantic concepts such as the shape of an instrument are expressed by the Murrinh-Patha classifier stems and it is these semantic concepts that account for the compatibility of the combinations of Murrinh-Patha classifier and lexical stems. The Murrinh-Patha data thus suggests that information which is normally considered encyclopedic knowledge has been grammaticalized in Murrinh-Patha classifier stems.
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Seiss, Melanie. 2011. Implementing the morphology-syntax interface: Challenges from Murrinh-Patha verbs. In Miriam Butt & Tracy Halloway King


Street, Chester. 1989. Murrinh-Patha vocabulary, electronic ms word file. Ms, Darwin N.T.


