Causatives of transitives: cross-linguistic analysis and implementation

Inaugural-Dissertation
zur Erlangung des Doktorgrades der Philosophie (Dr. phil.)
durch die Philosophische Fakultat
der Heinrich-Heine-Universität Düsseldorf
vorgelegt von

Valeria Generalova

Betreuerin: Laura Kallmeyer
Betreuer: Robert D. Van Valin, Jr.

Düsseldorf, August 2023

Лизе,
навсегда задавшей самые высокие планки,

и Ленинградской типологической школе,
научившей прыгать

Causatives of transitives: cross-linguistic analysis and implementation

Valeria Generalova

Abstract

Languages differ with respect to the number of arguments they allow in a simple clause. Two-argument clauses (a subject and a direct object) are attested in the vast majority of languages, whereas languages allowing three-argument cores are less frequent. Hence, adding an argument to an intransitive verb would yield a two-argument construction, which would probably look like a prototypical transitive construction attested in this language. However, adding an argument to a transitive verb is not that straightforward.

This thesis investigates the argument structure of constructions with causative verbs morphologically derived from transitive bases from the perspective of Role and Reference Grammar (Van Valin 2005; Van Valin 2023b). The main method of re-analyzing causative constructions in each language is relating them not to the constructions they are derived from but to other three-argument patterns attested in the same language (if there are any). In the absence of three-argument patterns, two-argument constructions featuring the same means of argument marking as in the constructions in question are considered. The method is inspired by Kemmer and Verhagen (1994) and developed to suit structurally varied languages. As a result, the thesis offers a functionally motivated enumeration of argument-marking strategies in causatives-of-transitives used by different languages. Language-related observations are formulated in the spirit of comparative concepts (Haspelmath 2010) to be compatible with further cross-theoretical studies. Construction-related observations are presented as Constructional Schemas, contributing to the classical RRG and to the RRG-inspired implementation presented in the second part of the thesis.

The suggested implementation relies on the formalization of RRG as a Tree-Wrapping Grammar (Kallmeyer and Osswald 2023) and uses XMG-2 (Crabbé et al. 2013; Petitjean, Duchier, and Parmentier 2016). This is the most multilingual XMG project to date, and it can be easily expanded to cover more languages and other grammatical phenomena.

Acknowledgements

This thesis was realized as part of the TreeGraSP project funded by an ERC Consolidator Grant awarded to Prof. Dr. Laura Kallmeyer. Therefore, first and foremost, I would like to thank her for the opportunity to come to Düsseldorf and do computational RRG. And for a quick reaction to every really urgent issue. And for extending my working contract several times.

Second, I would like to thank Van – also known as Robert D. Van Valin Jr. – for taking me as a student notwithstanding his retirement, for trusting in me and my project, and for offering exhaustive answers to any question. And for creating RRG, first of all. And for reminding me that "there is life after PhD".

Rainer Osswald was not supposed to be my official supervisor, but he joined at a very early stage and remained active until the very end of the work. He was always there to point out the weaknesses of my analysis, to ask questions I did not dare ask myself, and to put the answers of my official supervisors in other words. I am extremely grateful to him for that.

A significant part of my project requires using XMG, and I thank Simon Petitjean for making me enjoy this language. I really treasure this addition to the list of languages I master. And who fixed the compilers and the parser every time I crushed them. And again. And who proofread half of my thesis.

I am very grateful to the people who shared their fieldnotes and unpublished ideas on some of the languages I worked with: Elena Perekhval'skaya, Anna Smetina, Peter Arkadiev, Polina Oskol'skaya, Jozina Vander Klok, Kyle Jerro. They made the process of acquiring missing language data unbelievably quick and smooth. And pleasant.

Speaking of pleasant things, I must thank my dearest friend Dasha Kondakova – turned Dr. Daria Kohler while this thesis was being written – for the immense positive effect she has on my (academic ∪ ¬academic) life. And for countless little (but by no means minor!) effects she had on the text of this thesis.

Finally, I want to thank my mother for making this life-long journey possible and meaningful.

Let this be an expression of my love for science and those who taught me this love.

Abbreviations

□	adapted translation
△	adapted glossing
1	first person
2	second person
3	third person
A	actor
ABL	ablative
ABS	absolutive
ACC	accusative
ACT	active
ADESS	adessive
ADJ	adjective
AFFMT	affirmative
ALL	allative
APPL	applicative
AUX	auxiliary
BEN	benefactive
C1	noun class 1
C10	noun class 10
C11	noun class 11
C12	noun class 12
C15	noun class 15
C1A	noun class 1a
C3	noun class 3
C5	noun class 5
C6	noun class 6
C7	noun class 7
C9	noun class 9
CAUS	causative
CVB	converb
DAT	dative
DEF	definite
DET	determiner
ERG	ergative
EXT	stem extension
F	feminine
FIN	finite
FOC	focus
FUT	future
FV	final vowel
GEN	genitive
IND	indicative
INDF	indefinite
INF	infinitive
INS	instrumental
IO	indirect object
IPFV	imperfective
LOC	locative
M	masculine
N	neuter
NEG	negative
NMLZ	nominalizer
NMR	non-macrorole participant
NOM	nominative
OBJ	object
OBL	oblique
PA	pronominal anaphor
PAR	partitive
PART	particle
PASS	passive
PFV	perfective
PL	plural
POSS	possessive
PRED	predicative
PRF	perfect
PROG	progressive
PRS	present
PSA	privileged syntactic argument
PST	past
PSTPFV	past perfective
PTCP	participle
PURP	purposive
RCPST	recent past
REFL	reflexive
RTL	re-telling
SBJ	subject
SG	singular
ST	derived stem
TNS	tense
TV	thematic vowel
UG	undergoer

Index of Languages

Abkhaz-Adyghe viii, 136, 138, 139
  Circassian 141, 142
    Adyghe (West Circassian) ADY adyg1241 ix, x, xii, 138–143, 229, 244, 291
    Kabardian (East Circassian) KBD kaba1278 viii, xvii, 70, 136–143, 159, 166, 208, 210, 241
Altaic 96, 99, 159, 161, 163, 170, 238
  Mongolic 79, 96
    Kalmyk XAL kalm1243 x, xvii, 32, 79, 96, 98, 99, 112, 159, 162, 163, 167, 291, 293
    Khalkha Mongolian KHK halh1238 57
    Mongolian MON mong1331 78, 79
  Tungusic 96
  Turkic 79, 84, 94, 96, 160, 162, 163, 238
    Bashkir BAK bash1264 viii, x, xii, xvi, 73, 79, 80, 84, 94–101, 105, 107, 112, 116, 117, 148, 167, 170, 246, 248, 291, 292
    Khakas KJH khak1248 xvii, 159–161, 167, 292
    Mishar Tatar (West Tatar) west2405 79, 95, 98, 99, 292
    Turkish TUR nucl1301 31, 72, 292
    Tuvinian (Tuvan) TYV tuvi1240 viii, xii, xvii, 159–161, 163, 167, 211, 212, 292
Arawakan
  Piro (Purus) PIB puru1265 29
Austronesian 63, 72, 104
  Chamorro CHA cham1312 63
  Javanese JAV java1254 x, xvi, 72, 104–107, 166, 262
  Sama sama1302 22
Bantoid
  Bantu 13, 18, 31, 41–43, 63, 71, 72, 100, 117–122, 125, 128–131, 135, 161, 167, 188
    Chichewa (Chewa) chew1246 41, 42
    Ikota (Kota) KOQ kota1274 188, 261
    Kikuyu (Gikuyu) KIK kiku1240 viii, x, xvii, 13, 130–136, 167
    Kinyarwanda KIN kiny1244 xvii, 13, 118, 129–136, 167

Lubukusu (Bukusu) BXK buku1249 .. xvii, 100, 105, 119, 120, 122, 123, 125–129, 148, 167, 170, 291
Swahili SWK swah1253 72
Zulu ZUL zulu1248 .... viii, xvii, 123–126, 128, 167, 262
Wide Grassfields ....
Babungo (Vengo) BAV veng1238 72
Barbacoan ....
Awa Pit Pit (Awa-Cuaiquer) KWI awac1239 262
Dravidian 69, 84, 86
Kannada KAN nucl1305 .... xvi, 47, 78–80, 84, 86–90, 93, 94, 107, 112, 291, 292
Telugu TEL telu1262 .... xiii, 69, 209, 248–250, 292
Eskimo-Aleut ....
Asiatic Eskimo (Central Siberian Yupik) ESS cent2128 .... 291, 292
Greenlandic Eskimo (Kalaallisut) KAL kala1399 .... viii, 135
Heibanic (Kordofanian) 121
Moro MOR moro1285 121
Indo-European 69
Germanic 183, 258
Dutch NLD dutc1256 .... 57, 78, 184
English ENG stan1293 ..... ix, x, 4, 8, 13, 15, 21, 22, 30, 34, 44, 61, 86, 88, 89, 109, 141, 148, 161, 179, 181, 187, 188, 207, 211, 212, 215, 257, 259
German DEU stan1295 .... 13, 61, 77–80, 184, 258
Icelandic ISL icel1247 .... 22, 23, 122, 123
Indo-Aryan ....
Hindi HIN hind1269 .... 45, 59
Sanskrit SAN sans1269 46
Sinhalese (Sinhala) SIN sinh1246 .... 69, 70
Urdu URD urdu1245 59
Italic ....
Latin LAT lati1261 .... 46, 130, 133–135, 141, 209
Western Romance 72
Slavic ....
Croatian HBS sout1528 18
Russian RUS russ1263 .... viii, 27, 57, 74, 95, 96, 133, 134, 161, 162, 164, 183, 188, 261
Isolates ....
Nivkh (Gilyak) NIV gily1242 .... x, xvi, 36, 69, 74, 112–117, 166, 208, 292
Trumai TPY trum1247 .... viii, x, xi, xvii, 70, 136, 137, 144–159, 166, 241
Japonic ....

Japanese JPN nucl1643 . . . . 22, 48, 90, 91, 162, 262
Koreanic . . . .
Korean KOR kore1280 57
Mayan . . . .
Tzotzil TZO tzot1259 13
Nakh-Daghestanian 89
Daghestanian . . . .
Tsez (Dido) DDO dido1241 262
Nakh . . . .
Tsova-Tush (Batsbi, Bats) BBL bats1242 89
Pama-Nyungan . . . .
Dyirbal DBL dyir1250 . . . . 4, 148
Quechua 63, 78
Bolivian Quechua boli1262 . . . . 90, 93
Salish 13
Halkomelem HUR halk1245 13
Semitic . . . .
Hebrew HEB hebr1245 47
Sino-Tibetan . . . .
Mandarin Chinese CMN 183
Qiang qian1264 70
Siouan 11
Lakhota (Lakota) LKT lako1247 . . . . viii, 4, 11–13, 262
Uralic 107
Finnic . . . .
Estonian EKK esto1258 . . . . 84, 85, 291
Finnish FIN finn1318 . . . . viii–x, xii, xvi, 58, 73, 74, 76, 79, 80, 107–112, 128, 166, 167, 190, 223, 224, 237, 244, 255, 256, 291
Hungaric . . . .
Hungarian HUN hung1274 . . . . 49, 50
Uto-Aztecan . . . .
Yaqui YAQ yaqu1251 . . . . 255, 292, 293
Wakashan . . . .
Nootka (Nuu-chah-nulth) NUK nuuc1236 10

List of Figures

1 The organization of RRG (reproduced from Van Valin 2023b, 19) 6
2 The sample layered structure of the sentence 9
3 Two traditions for representing peripheral elements 10
4 The layered structure of Lakhota sentences (1) and (2) (reproduced from Van Valin 2005, 147, 148 with modifications) 12
5 Examples of nexus-juncture types 20

6 Causative situation (reproduced from Nedyalkov and Sil'nitskiy 1969a, 6) 28
7 Types of causative opposition (according to Nedyalkov and Sil'nitskiy 1969b, 20–22) 29

8 An updated algorithm of causee coding depending on implicativeness and volitionality 92
9 Constructional Schemas for Bashkir causatives 101
10 Causative strategies with respect to causative and ditransitive alignments 102
11 The syntactic representation of Finnish causative-of-transitive (57) as a two-argument core 111
12 The tree representation of a Zulu causative (77a) 124
13 The tree representation of the Kikuyu ditransitive (86) (reproduced from Kihara 2017, 60, Fig. 3.2) 132
14 A Russian sentence without an overt subject NP (taken from RRGParBank) 134
15 A classic RRG analysis of noun incorporation in Greenlandic Eskimo . . 135
16 The genealogical structure of the Abkhaz-Adyghe family[1](Hammarström et al. 2022; Eberhard, Simons, and Fennig 2021) 139
17 The suggested syntactic analysis of Kabardian causative constructions . . 143
18 The analysis of Trumai causative (98) 'Alaweru made me beat the child.' as core subordination 154
19 Syntactic descriptions of Trumai causative constructions 157
20 A dependency syntactic analysis of the Russian causative stimulus (115) 161
21 The tree representation of the Tuvinian causative with an "incorporated" affectee (114) 163

22 Parsing operations (reproduced from Kallmeyer and Osswald 2023, Fig. 20.7, Fig. 20.11, modified, simplified) 177
23 The frame represenation of the English verb *break* regarded as a causative (reproduced from Kallmeyer and Osswald 2013, p. 279, Fig. 6a) 181
24 The frame used for causatives-of-transitive 182
25 Examples of trees generated by the classical XMG (reproduced from Crabbé et al. 2013, p. 595, Fig. 2) 185
26 The output representation for the Finnish sentence 'Satu made Diane kill the slug.' 190
27 The setup for the suggested solution 191

28 The internal structure of an XMG class 193
29 The model generated by Snippet 1 195
30 Technical limitations of properties 199

31 Example (21) as source LaTeXcode 210
32 Anchoring inflectional morphemes 221
33 Models generated by the class `NounWithCase` and the corresponding Finnish words 223
34 The elementary tree for arguments produced by the class `Noun With Case` (with the class `Nominal Spine` imported) 225
35 The non-lexicalized model produced by the class `Nominal Spine` (without being imported elsewhere) 226
36 An alternative import of the class `Noun With Case` into the class `Nominal Spine` 227
37 The derived tree for the Adyghe sentence 'Mother made the girl write a letter.' 229
38 Tree representations produced by the `class ArgumentHM` (feature structure simplified) 230
39 Outputs of the class `VerbalRoot` for transitive morphemes 232
40 The derivation of the tree for the Finnish verb *tapa-tt-i* 237

List of Tables

1 Verb classes (according to Van Valin 2023b, p. 99, (91)) 7
2 The Actor-Undergoer Hierarchy (reproduced from Van Valin 2023b, p. 108, (99)) 15
3 Semantic relations conveyed by all nexus-juncture types with English examples (modified from Van Valin 2023b) 21
4 Types of causatives (according to Van Valin 2023b, 84–86) 21
5 Modules of RRG and their outputs 25
6 Types of causative constructions (according to Song 1996, Fig. 5.2, p. 138; reproduced with modifications) 35
7 The Paradigmatic Argument Selection Principle (reproduced from Ackerman and Moore 1999, p. 15, (24)) 39
8 Argument patterns attested with verbs of transfer (reproduced from Anderson 2005, p. 50, (45a, 45b)) 44
9 Causative of transitive types (reproduced from Dixon 2000, p. 48, (24)) . 68
10 Variation in case marking on causee of causative-of-transitive (reproduced with minor modifications from Kemmer and Verhagen 1994, 137) 78
11 Alternations in causee coding in causative constructions (reproduced with some adjustment from Van Valin 2018a, 127) 90
12 The feature ‘benefit’ in Bashkir causative constructions 100
13 Relation of type **Affectee like Recipient** between (51) and (49) in Javanese 105
14 Agent, Demoted agent, and Instrument compared 111
15 Marking of causees, demoted agents and instruments in Finnish, Bashkir and Kalmyk 112
16 Constructional Schema for Nivkh causatives 117
17 Properties of arguments in Kikuyu ditransitive constructions 133
18 Basic bivalent and trivalent verbal alignments in Adyghe (from Smeets 1992, 127; modified) 140
19 Causative derivation in West Circassian (Smeets (1992, 127), modified) . 142
20 Cases used for encoding core arguments in Trumai 146

21 Traditional interpretations of the Trumai verbal system 149
22 Trumai bivalent patterns and their causative counterparts 151
23 Relation of type **AffecteeLikeRecip** between (104) and (102) in Trumai 152

24 The range of causative constructions attested in sample languages 169
25 Constructional schemas for causative constructions 172

26 Feature unification 198
27 Values of the property `mark` 199

List of Snippets

1 An example class illustratung the frame dimension (taken from the XMG documentation) . . . 195
2 Excerpts from the preamble illustrating different kinds of features 197
3 Atomic disjunction in the absence of negation . . . 201
4 Implication presented as disjunction of conjunctions . . . 203
5 Example of using the dot notation . . . 203
6 Using the dot notation for accessing language-specific information . . . 204
7 The `morpho` class showing the Tuvinian ACC marker . . . 211
8 The lemma class for a causative affix . . . 215
9 Values of `type CAT` given in the preamble . . . 216
10 Abstractions for Lemma classes . . . 219
11 Importing an abstraction and specifying the entry of a Lemma class . . . 219
12 The class for anchoring a single morpheme . . . 220
13 The class describing nominals . . . 221
14 The class describing the wrapping substitution of arguments in head-marking languages . . . 228
15 The class describing verbal roots . . . 231
16 The class describing causative verbal stems . . . 233
17 The class modeling the valency increase . . . 235
18 An extension of the class `ValIncrease` blocking four-argument cores 235
19 The class describing the verbal spine . . . 236
20 The class describing the placement of the object immediately before the verb . . . 238
21 The class describing all possible causative templates . . . 239
22 The class specifying argument coding in causatives of transitives . . . 241
23 The class describing causative templates in head-marking languages . . . 243
24 Intersection classes (exemplified by Finnish and Adyghe) . . . 244
25 The language plugin for Bashkir . . . 246
26 The usage of `?LanguageFeatures` in Construction Classes and Language Plugins . . . 247
27 Case features defined in the preamble . . . 247
28 Inventory booleans in Bashkir . . . 248

29 The `morpho` class for a case marker realized as morphological zero . . . 249
30 Unmarked cases in Telugu (fragment of a language plugin) 250
31 A language plugin in action: some examples 250

Contents

Abbreviations **iv**

Index of Languages **v**

List of Figures **viii**

List of Tables **x**

List of Snippets **xii**

1 Introduction **1**

I Review part **3**

2 Some relevant concepts of Role and Reference Grammar **4**

2.1 Reference sources . . . 5
2.2 General architecture . . . 6
2.3 Semantic representations . . . 7
2.4 The layered structure of the sentence . . . 8
2.4.1 The layered structure in dependent-marking languages . . . 8
2.4.2 The layered structure of the phrase . . . 9
2.4.3 A terminological note on RP vs. NP . . . 10
2.4.4 The operator projection . . . 11
2.4.5 The layered structure in head-marking languages . . . 11
2.5 Arguments . . . 13
2.5.1 Direct vs. oblique arguments vs. adjuncts . . . 13
2.5.2 Macroroles . . . 14
2.5.3 Case assignment . . . 17
2.5.4 Closing remarks on arguments in RRG . . . 19
2.6 Juncture-nexus types . . . 20
2.6.1 General presentation . . . 20
2.6.2 Causative constructions as nexus juncture types in classical RRG . . . 21
2.7 Constructional schemas . . . 22

2.8 The linking algorithm . . . 24

3 Prior approaches to (three-argument morphological) causatives 26
3.1 Introduction and motivation . . . 26
3.2 Xolodovič (1969): the first typological volume . . . 27
3.3 Comrie (1976): the most influential study . . . 30
3.4 Replies to Comrie (1976) . . . 33
3.4.1 Song (1996) on NP density control . . . 33
3.4.2 Li (2008) critisizing Comrie (1976) and Song (1996) . . . 36
3.5 Inspired by Dowty (1991) . . . 38
3.5.1 Ackerman and Moore (1999): ingredients of the analysis . . . 38
3.5.2 Li (2020): weighted proto-properties . . . 40
3.6 Studies of the causee . . . 41
3.6.1 Alsina (1992): attention towards the argument structure . . . 41
3.6.2 DeLancey (1984): an early prototype approach to features . . . 43
3.6.3 Anderson (2005): features instead of grammatical relations . . . 44
3.6.4 Saksena (1980): affectedness matters . . . 45
3.6.5 Wali (1981): causees as second-order agents . . . 46
3.6.6 Cole (1983): causees retain their agentivity . . . 47
3.6.7 Horvath and Siloni (2011): a particular approach to demotion . . . 48
3.7 Kemmer and Verhagen (1994): situating causatives in a language . . . 51
3.8 Shibatani and Pardeshi (2002) and De Busser (2010): features and dimensions 55
3.9 Kozinsky and Polinsky (1993) on coding conflict . . . 56
3.10 Selected generative studies . . . 58
3.10.1 Pylkkänen (2008) on the addition of arguments . . . 58
3.10.2 Copley and Martin (2014a) on the cognitive dimension . . . 59
3.10.2.1 Copley and Wolff (2014) . . . 59
3.10.2.2 Martin and Schäfer (2014) . . . 61
3.10.3 Hoffman (1991) on accusative vs. oblique causee languages . . . 62

II Theoretical part 65

4 Introduction 66

5 Background and method 68
5.1 Dixon (2000): a starting point . . . 68
5.1.1 Type (i) . . . 69
5.1.2 Type (ii) . . . 70
5.1.3 Type (iii) . . . 70

5.1.4 Type (iv) 71
5.1.5 Type (v) 72
5.2 Criticism of Dixon (2000) summarized 73
5.3 Inspiration from Kemmer and Verhagen (1994) 75
5.4 My approach illustrated 81

6 Argument structure and linking in each strategy **84**
6.1 Strategies without doubling 84
6.1.1 Causee like Recipient & Causee like Demoted Agent 84
6.1.1.1 Strategies presented 84
6.1.1.1.1 Causee like Recipient 84
6.1.1.1.2 Causee like Demoted Agent 86
6.1.1.1.3 Data: Kannada 86
6.1.1.2 Analysis 1: The volitionality hypothesis 87
6.1.1.3 Analysis 2: The Agentivity Implicature hypothesis 89
6.1.1.4 Bashkir 94
6.1.1.4.1 New data 94
6.1.1.4.2 An alternative analysis 96
6.1.1.5 Summary 100
6.1.2 Other ditransitive-like strategies 101
6.1.2.1 Description of the strategies 101
6.1.2.2 Data: Javanese 104
6.1.2.2.1 General concerns 104
6.1.2.2.2 Possible causative interpretation 105
6.1.2.2.3 Type **Causee like Recipient** in Javanese 106
6.1.2.3 Summary 107
6.1.3 Causee like Instrument 107
6.1.3.1 Data: Finnish 107
6.1.3.2 A prior non-RRG analysis 109
6.1.3.3 An RRG analysis 110
6.1.3.4 Discussion 111
6.1.3.5 Summary 112
6.1.4 Causee like Agent of a Converbial Clause 112
6.1.4.1 Data: Nivkh 113
6.1.4.2 Relatable constructions 114
6.1.4.3 The suggested RRG analysis 116
6.1.4.4 Summary 117
6.2 Strategies with doubling 117
6.2.1 Double objects 117

6.2.1.1 General notes . . . 117
6.2.1.2 Challenge 1: Object (a)symmetry . . . 119
6.2.1.2.1 Key notions . . . 119
6.2.1.2.2 A prior HPSG analysis . . . 120
6.2.1.2.3 An RRG analysis . . . 121
6.2.1.2.4 Zulu: Data and analysis . . . 123
6.2.1.2.5 Lubukusu: Two patterns . . . 125
6.2.1.2.6 Summary . . . 128
6.2.1.3 Challenge 2: Locus of marking . . . 129
6.2.1.3.1 Kinyarwanda: Data and background . . . 129
6.2.1.3.2 Kikuyu: Data and a prior RRG analysis . . . 130
6.2.1.3.3 A modified RRG analysis . . . 132
6.2.1.3.4 Summary . . . 135
6.2.2 Candidates for double subjects . . . 136
6.2.2.1 General concerns . . . 136
6.2.2.2 Kabardian . . . 137
6.2.2.2.1 Introduction . . . 137
6.2.2.2.2 The slot structure of the verb . . . 138
6.2.2.2.3 Case marking . . . 140
6.2.2.2.4 Causative verbs . . . 142
6.2.2.2.5 Causative constructions: analysis and discussion . . . 142
6.2.2.3 Trumai . . . 144
6.2.2.3.1 Case marking in Trumai . . . 146
6.2.2.3.2 Two-argument patterns in Trumai . . . 147
6.2.2.3.2.1 A possible RRG analysis of Trumai bivalent patterns . . . 149
6.2.2.3.3 Three-argument non-causative patterns in Trumai . . . 150
6.2.2.3.3.1 A still possible RRG analysis . . . 150
6.2.2.3.4 Three-argument causatives in Trumai . . . 151
6.2.2.3.4.1 Background on the domain of case assignment . . . 152
6.2.2.3.4.2 An analysis of Trumai causatives inspired by the domain of case assignment . . . 153
6.2.2.3.4.3 Problematic RRG analysis as a three-argument clause . . . 155
6.2.2.3.4.4 An RRG analysis using oblique core arguments . . . 156
6.2.2.3.5 Conclusion . . . 158
6.2.2.4 Final remarks on "double subject" strategies . . . 159
6.2.3 Strategies with differential object marking . . . 159
6.2.3.1 Data: Kalmyk, Khakas, Tuvinian . . . 159
6.2.3.2 Prior analyses: language contact . . . 161
6.2.3.3 Prior analyses: incorporation . . . 162

6.2.3.4 An RRG analysis . . . 162
6.2.3.5 Summary . . . 163

7 From individual analyses to constructional schemas **165**
7.1 Summary of observations concerning individual languages . . . 165
7.2 Language profiles . . . 167
7.3 Summary of observations concerning cross-linguistic generalizations . . . 170
7.4 Constructional schemas . . . 171

III Implementation **173**

8 From theoretical abstractions to tangible representations **174**
8.1 Introduction and motivation . . . 174
8.2 From classical RRG to a formalized tree-wrapping grammar . . . 175
8.2.1 Background . . . 175
8.2.2 Operations in TWG . . . 176
8.2.3 Frame semantics for RRG . . . 179
8.3 TuLiPa: a parser for TWGs . . . 182
8.4 Metagrammar: describing language descriptions . . . 183
8.5 XMG: describing tree-based grammars with constraints . . . 184
8.6 Summary: theoretical and technical setup . . . 188

9 Key concepts of XMG **192**
9.1 The internal structure of the class . . . 192
9.1.1 Import . . . 192
9.1.2 Export . . . 193
9.1.3 Declare . . . 194
9.1.4 Dimensions . . . 194
9.1.5 Variables . . . 196
9.1.6 Nodes and edges . . . 197
9.1.7 Closing remarks . . . 197
9.2 Features . . . 197
9.3 Properties . . . 199
9.4 Logical operations . . . 200
9.4.1 Conjunction . . . 200
9.4.2 Logical disjunction . . . 200
9.4.3 Atomic disjunction . . . 200
9.4.4 Negation . . . 201
9.4.5 Implication . . . 202

9.5 Dot notation 203

10 The metagrammar of three-argument constructions with a morphological causative **206**

10.1 Overview 206
10.2 Data preparation 207
10.2.1 Sentence segmentation 208
10.2.2 Conversion from glossed examples to `.mg` files 209
10.3 Lexical Resources 211
10.3.1 Morpho 211
10.3.1.1 Morph 211
10.3.1.2 Lemma 212
10.3.1.3 Special features 212
10.3.1.3.1 Case affixes: feature `case` 212
10.3.1.3.2 Verbal roots: feature `trans` 212
10.3.1.3.3 Causative affix: features `caus` and `valtype` 213
10.3.1.4 Output 214
10.3.2 Lemma 214
10.3.2.1 Entry 215
10.3.2.2 Category 215
10.3.2.3 Family 216
10.3.2.4 Final remarks 218
10.4 Construction classes 218
10.4.1 Anchoring 219
10.4.2 Nominals 222
10.4.2.1 Noun with case 222
10.4.2.2 Nominal inflection 224
10.4.2.3 Nominal Spine 224
10.4.2.4 Argument in head-marking languages 227
10.4.3 Verbs 230
10.4.3.1 Verbal Root 230
10.4.3.2 Verbal Stem 232
10.4.3.3 Verbal inflection 234
10.4.3.4 Verbal Derivation: valency increase 235
10.4.3.5 Verbal Spine 235
10.4.4 Word order rules 237
10.4.5 Constructions 238
10.4.5.1 Dependent-marking languages 239
10.4.5.1.1 Generating all possible causative constructions 239

10.4.5.1.2 Specifying the argument encoding . . . 241
10.4.5.2 Head-marking languages . . . 243
10.4.6 Intersections . . . 244
10.5 Language Plugins . . . 245
10.5.1 Description . . . 245
10.5.2 Unmarked cases . . . 248
10.5.3 Grouping language plugins . . . 251
10.5.4 Discussion . . . 251

11 Evaluation and expansion **254**
11.1 Evaluating my solution . . . 254
11.2 Reproducing the results . . . 256
11.3 Comparison to other projects . . . 257
11.3.1 Early multilingual metagrammars . . . 257
11.3.2 FunGramKB + Artemis . . . 258
11.3.3 LinGO Grammar Matrix . . . 260

12 Summary **265**
12.1 Summary of computational contributions . . . 265
12.2 Final remarks and further questions . . . 266

A The code for the main MetaGrammar **268**

B Sample lexical metagrammars (example language: Finnish) **282**
B.1 Morpho . . . 282
B.2 Lemma . . . 287

C Testing results **291**

D Script for creating XMG lexical files from glossed examples **294**

References **305**

Chapter 1

Introduction

I will not start my thesis with the sentence "Causative constructions have played a crucial role in the development of linguistic theories". Instead, I start with the question: "Has anyone ever asked themself, what would a language do once forced to squeeze an additional subject into an already complete sentence?"

This thesis addresses this question by examining causative constructions morphologically derived from transitive verbs across structurally varied languages.

Causative is a well-known type of verbal derivation that adds an argument to a specific position. A handbook article by Haspelmath and Müller-Bardey (2004) offers the following definition of causative: "conveys the meaning of causation and adds a new agent argument (the causer) to the valence pattern". The authors also note that "the causer always usurps the subject function". A similar definition is found in Dixon (2000, 30): "a causative construction involves the specification of an additional argument, a causer, onto a basic clause". The same source also specifies that "a causer refers to someone or something (which can be an event or state) that initiates or controls the activity". Thus, the causative derivation is the most suitable terrain to study the phenomenon of adding an argument. Compared to an applicative construction, which also increases the valency of the verb, it always adds a new argument (while the status of "applied objects" is unclear in some languages).

Languages differ with respect to the number of arguments they allow in a simple clause. (For the time being, I will leave these terms in their intuitive or pre-theoretical understanding. More details on how I use them in my work are given in Chapter 2.) Two-argument clauses (a subject + a direct object) are attested in the vast majority of languages, whereas languages allowing three-argument cores are less frequent. Hence, adding an argument to an intransitive verb would yield a two-argument construction, which would probably look like a prototypical transitive construction attested in this language. However, adding an argument to a transitive verb is not that straightforward. Would a language allow a three-argument causative construction at all? If so, what other construction would it look like? If not, what would happen with the participants of the verbs? The present thesis seeks to give answers to all these questions.

The thesis deals exclusively with morphological causatives, since they are the most transparent structures reflecting the addition of an argument resulting in a simple clause. In lexical causatives, the *cause* and the *effect* are fused in a single morpheme. Hence, it is often unclear whether the argument structure of lexical causatives results from causative derivation or some other lexical idiosyncrasy. On the other hand, in syntactic (analytic, periphrastic) causatives, one can clearly separate the base verb and its arguments from the cause and the argument it introduces. Nevertheless, these causatives are often structures of a different kind than simple clauses, therefore, the addition of an argument follows different rules. It is important to mention that some morphological causatives are known to behave like syntactic ones. These cases are not included in the present study.

This thesis is organized as follows. It opens by explaining some key notions of Role and Reference Grammar to facilitate understanding of other chapters. This overview in Chapter 2 is not intended as a full presentation of RRG and does not aim to be one. Instead, it offers the option of reading other chapters to readers without prior experience with RRG studies.

Chapter 3 presents previous studies. In addition to presenting ideas that influence the reasoning in this thesis, it also showcases some other studies that might be known to future readers. With that, Chapter 3 aims to help people familiar with other theoretical approaches read an RRG-based study. Part II offers a theoretical RRG-based analysis of constructions with morphological causatives derived from transitive verbs. Chapter 5 explains the method used for analyzing the constructions in question. It starts by presenting and criticizing the typology of causatives-of-transitive suggested in Dixon (2000). Afterward, it presents the method of relating causatives to other three-argument constructions suggested in Kemmer and Verhagen (1994) and expands it. Finally, it outlines the method I use to enumerate the various strategies of argument marking in the constructions in question. Chapter 6 presents my analysis of various languages. More details on its internal structure can be found in the due sections. The part concludes with Chapter 7, which summarizes the theoretical findings and presents them in a suitable format for implementation. The motivations for implementation and the necessary background are provided in Chapter 8. The next chapter, Chapter 9, presents the notions necessary for understanding the details of the solution presented in Chapter 10. Chapter 10 is organized as a documentation of the solution, showing not only how the theoretical findings are implemented but also why each architectural decision was made. Chapter 11 offers the results of the tests and provides instructions necessary for reproducing these results and expanding the solution to cover more data. Chapter 12.1 summarizes the contributions of Chapter 10. Chapter 12.2 concludes the whole thesis and outlines the directions for further research. The thesis also contains several appendices for the readers' convenience.

Part I

Review part

Chapter 2

Some relevant concepts of Role and Reference Grammar

Role and Reference Grammar (RRG) has been developed, focusing on typologically diverse languages. The initial question that RRG addresses is:

> What would linguistic theory look like if it were based on the analysis of languages with diverse structures, such as Lakhota, Tagalog, and Dyirbal, rather than on the analysis of English? Van Valin (2005, 1)

As a result, RRG redefines some fundamental linguistic concepts (e.g., *subject*) to adequately describe structures that differ significantly from those found in commonly studied languages, particularly English.

The second crucial question posed by RRG goes a step further: it not only seeks to analyze language structures but also to explore their interrelations.

> How can the interaction of syntax, semantics, and pragmatics in different grammatical systems best be captured and explained?
> Van Valin (2005, 1)

Addressing this question shapes the functionalist perspective of RRG and has led to significant findings, including the syntax $\leftrightarrow$ semantics linking algorithm and the concept of constructional schemas, both of which are essential for the present work.

The primary purpose of this chapter is to introduce key concepts of RRG that are necessary for understanding the remainder of the thesis. I intend to make this thesis accessible not only to those already familiar with RRG or interested in it but also to provide explanations of specific RRG concepts using theory-independent reasoning. While it may seem unfair to present the theory underpinning my work without a critical evaluation of its overall effectiveness and capacity to describe various linguistic phenomena, I must balance brevity and readability for my readers. Unfortunately, it is difficult to find evaluations of RRG written by RRG-external scholars because there is generally little cross-theoretical research in linguistics nowadays. Nonetheless, I will take a step back to position RRG ideas within the broader landscape of linguistic knowledge.

2.1 Reference sources

The first volume presenting RRG is "Functional syntax and universal grammar" (Foley and Van Valin 1984). Its main aim is to highlight the areas where generative studies are not powerful enough and to justify the right to reconsider fundamental linguistic notions. The book consists of seven chapters, each following the same conceptual outline: first, typological evidence is presented; then, discussion and analysis follow; and then the chapter concludes with an expansion of the suggested analysis. The concepts introduced in the first six chapters were later reanalyzed and reorganized in subsequent versions of RRG. The seventh chapter provides an intriguing perspective on systems of discourse cohesion ("switch-reference" vs. "switch-function", see Foley and Van Valin (1984, 321–367)), which has not received much development since then.

There were two textbooks on RRG published in 1997 (Van Valin and LaPolla 1997) and in 2005 (Van Valin 2005), respectively. Both offer a systematic explanation of the theory and are frequently cited. For a clear, concise, and well-structured overview of RRG theory, one would use the 2005 book (Van Valin 2005). In contrast, those seeking more language data and detailed discussions of nuanced cases should consult the 1997 book (Van Valin and LaPolla 1997).

Curiously, neither of these books includes the term "Role and Reference Grammar" in their titles. According to Van Valin (p. c.), the term emerged in response to Schachter (1977), which was titled "Reference-related and role-related properties of subjects". The term *Role and Reference Grammar* first appeared in Van Valin (1977, 699) just *en passant*, without any grand announcement of a new theory. Initially associated with Van Valin's efforts to redefine grammatical relations, it has since evolved into a theory encompassing much more.

Recently before the submission of this thesis, "The Cambridge Handbook of Role and Reference Grammar" (Bentley et al. 2023) appeared. Featuring 26 contributions organized into four parts, it presents the latest principles of RRG (Van Valin 2023b) alongside research on simple and complex sentences. One section is dedicated to applications of RRG, among which Kallmeyer and Osswald (2023) is particularly relevant for this thesis. The final section includes RRG-driven grammatical sketches of various languages.

RRG is a dynamic and expanding theoretical framework with various adaptations and dialects. While this thesis adheres to the "classical" RRG, I would like to mention two of those dialects. The first one has originated from the work of Brian Nolan and Elke Diedrichsen, both of whom were colleagues at TU Dublin at one point (Nolan and

Diedrichsen 2013; Nolan, Rawoens, and Diedrichsen 2015; Nolan and Diedrichsen 2019). This approach to RRG is more construction-oriented than the classical version, advancing and refining the concept of constructional schemas (see Section 2.7). Another dialect worth mentioning is the “Radical Role and Reference Grammar (RRRG)” (Kailuweit 2013). Its main objective is to eliminate the traditional linking algorithm (see Section 2.8) and replace it with three independent layers: “lexical items, syntactic-semantic event templates and construction schemas” (Kailuweit 2013, 103). Although I did not base my study on Radical RRG, this general framework partly resembles the outline of my own implementation (see Chapter 10).

The references cited above represent just a fraction of the extensive literature on RRG. The RRG bibliography[2] is continually updated and includes numerous studies on various languages and phenomena. The sections below summarize key insights from the main RRG sources and from additional papers focused on specific topics.

2.2 General architecture

Syntactic representation

↑

Linking algorithm

↓

Semantic representation

Discourse-pragmatics

Fig. 1. The organization of RRG (reproduced from Van Valin 2023b, 19)

The general architecture of RRG consists of four interrelated modules, as illustrated in Fig. 1. Each module (syntax, semantics, discourse) has a different architecture to suit the nature of the entities it describes. The Linking Algorithm is a separate module.

The semantic representation is based on the theory of *Aktionsart* and is articulated through *logical structures*. While these representations often resemble structures found in predicate logic, they are far less formal. Section 2.3 discusses classical RRG semantic structures, while Section 8.2.3 presents critiques by Osswald and Kallmeyer (2018) along with their suggestion for a more formalized and machine-compatible data structure for semantic representation, which would still inherit the spirit and the overall structure from RRG.

The syntactic representation in RRG is distinct in that it moves away from traditional grammatical relations and introduces a new system of terms to characterize sentence components more precisely. However, RRG remains a tree-based grammar, although its

[2] https://rrg.caset.buffalo.edu/, accessed on 01.05.2026

trees differ significantly from the (generative-inspired, mostly binary) trees that many readers may be familiar with. Most of the remainder of the present chapter delves into various aspects of this RRG module.

The Linking Algorithm is a module that bi-directionally connects syntactic and semantic representations. In classical RRG, this algorithm is realized as a set of rules (commands), but recent computationally oriented works suggest a different formal architecture. I present it in rather general terms in Section 2.8 and re-model it computationally in Chapter 10.

The discourse-pragmatics module is involved in the linking process but is not part of the basic linking algorithm. It consists of *information structure* representations of sentences, which are specific to RRG. This module is of minor relevance to the present study; therefore, I do not provide a detailed review here. Nonetheless, I incorporate discourse-pragmatic reasoning in Section 6.1.1.3 and discuss some innovations in RRG within that context.

2.3 Semantic representations

Verb class	**English example**	**Semantic representation**
STATE	John saw the picture.	**see′** (John, picture)
ACTIVITY	The children cried	**do′** (children, [**cry′** (children)])
ACHIEVEMENT	The climber reached the summit.	INGR **be-at′** (summit, climber)
SEMELFACTIVE	Mary coughed.	SEML **do′** (Mary, [**cough′** (Mary)])
ACCOMPLISHMENT	Mary learned French.	BECOME **know′** (Mary, French)
ACTIVE ACCOMPLISHMENT	Chris ran two miles to the park.	**do′** (Chris, [**run′** (Chris)]) ∧ PROC **covering.path.distance′** (Chris, two miles) ∧ FIN **be-at′** (park, Chris)

Table 1: Verb classes (according to Van Valin 2023b, p. 99, (91))

The RRG approach to semantics is based on the *Aktionsart* theory (Vendler 1967) but revisits the list of event classes. One significant innovation of RRG is the introduction of the class of *active accomplishments*, which is described as “the telic use of activity verbs” (Van Valin 2005, 32). Additionally, RRG introduces a distinct class for semelfactives, which were not listed in Vendler (1967). The classification is still evolving, benefiting from ongoing debates about specific event classes.

Predicates and their arguments are represented as so-called *logical structures*. Lexical predicates are depicted as functions that take a bracketed array of arguments. Lexical predicates are written in boldface and followed by a prime, e.g., **see′**(x, y) or **sleep′**(x). A special shortcut is used to represent an unspecified activity: [**do′** (x, ∅)]. Square brackets have no specific function and are used alongside round brackets solely to enhance readability.

Certain types of events are written as capitalized "operators" (term used in Van Valin (2005)) or "functions" (term used in Van Valin (2023b)). However, they only specify the event type denoted by the following logical structure (without putting it in brackets). Therefore, I prefer to refer to them as *types*. The types utilized in Van Valin (2023b) are: INGRessive, SEMeLfactive, BECOME, PROCess, FINal stage (of a process), CAUSE. One can notice that these types are very different in nature and scope. Tab. 1 presents all the types of events along with English examples and their full representations. It is placed here primarily for reference. Comments on each structure I use appear in their proper place throughout this work.

As illustrated in Tab. 1, RRG does not have a single class for causative verbs. Instead, causation is seen as a semantic derivation that can apply to any class. Thus, the operator CAUSE can, in principle, unite any two logical structures. Typically, the first component (denoting the action of causation) is usually an unspecified activity predicate: [**do′** (x, ∅)]. However, any semantic nuances of causation can be encoded in this part of the logical structures (see some remarks in Van Valin (2023b, 99–100)).

2.4 The layered structure of the sentence

2.4.1 The layered structure in dependent-marking languages

The central component of the syntactic representation in RRG is the layered structure of the clause. This structure postulates multiple *layers* of the syntactic tree over which syntactic and semantic effects can scope. Scopes of a given effect must not be uniform in all languages.

All possible structures and their positions in a tree description are displayed in Fig. 2. It is important to note that not all these structures are typically found within a single language. For example, Pre- and Post- positive elements depend on word order and thus are in complementary distribution. (However, according to Van Valin, p.c., exceptions do exist. For instance, Japanese is claimed to have a Pre-Core Slot, a Post-Core Slot, a Pre-Determiner Position, and a Post-Determiner Position).

A fundamental element present in every sentence across all languages is the predicate. In RRG, it is represented in the NUCLEUS (NUC) layer. Whether the predicate is verbal or non-verbal, it always belongs to the NUC structure.

RRG defines the CORE layer as a structure containing the predicate and all its semantically obligatory arguments (i.e., arguments that appear in the semantic representation of the predicate). CORE arguments can be direct or oblique (see in Section 2.5.1). Non-

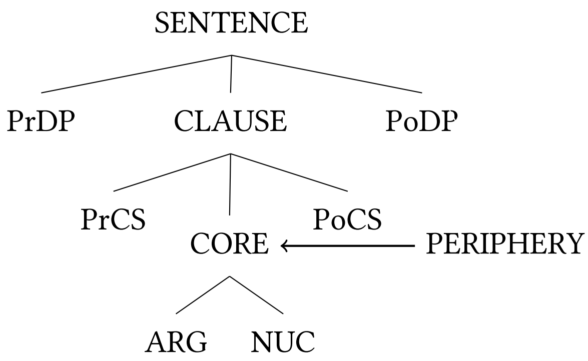


Fig. 2. The sample layered structure of the sentence

arguments, i.e., non-obligatory participants and circonstants, appear outside the CORE, i.e., in the PERIPHERY.

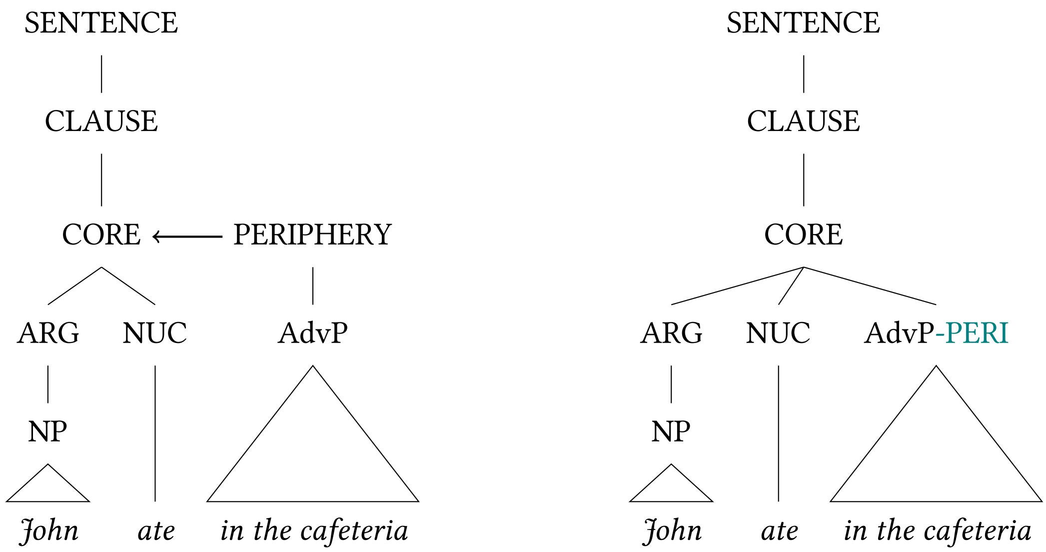
A short remark on the notation is necessary here. In classical RRG notation, as depicted in Fig. 2, PERIPHERY exists at the same hierarchical level as the CORE but is entirely unrelated to it and to other nodes in the same graph. However, this representation is somewhat misleading, as peripheral elements still belong to the same CLAUSE. Therefore, the newer notation (used in Bladier et al. (2022)) attaches peripheral elements to the main tree graph and assigns them a special tag, PERI. A comparison of the two notations is presented in Fig. 3.

The layer combining CORE and PERIPHERAL elements is the clause. There exists one special position within the CLAUSE that is neither part of the CORE nor the PERIPHERY: the PreCore Slot (or a PostCore Slot in verb-final languages), which hosts question words or topicalized elements.

Additionally, there is a layer above the clause: the SENTENCE. Apart from the clause, it can also contain a *detached position*, which is the place for sentence-initial or sentence-final elements that are "set off from the clause by a pause" (Van Valin 2005, 6). In classical RRG, they are called the *left-detached position* (LDP) and the *right-detached position* (RDP). However, it has been observed that the terms *left* and *right* have reverse meanings in right-to-left writing languages, making them typologically non-universal. The updated version of RRG annotation uses the terms *pre-detached position* (PrDP) and *post-detached position* (PoDP), which are independent of the direction of writing.

2.4.2 The layered structure of the phrase

It is essential in RRG that not only clauses but also all types of phrases maintain the same layered structure. In classical RRG (Van Valin 2005, 24–27), only the layered structure of NP – in terms of classical RRG, of RPs – is illustrated. In the newer tradition (Bladier



a. The classical notation

b. The new notation

Fig. 3. Two traditions for representing peripheral elements

et al. 2022), this principle is expanded so that all phrases – NP, PP, AdjP, AdvP – are recognized as having layered structures. In these constituents, the layer of *phrase* (e.g., NP) is parallel to the layer of the *clause*. Consequently, there are levels of CORE_N and NUC_N in an NP, along with the respective COREs and NUCs in other phrases. Peripheral elements and operators (see below) can be attached at each of these levels.

2.4.3 A terminological note on RP vs. NP

Classical RRG discards the traditional notion of NP and proposes the term *RP*, which stands for *reference phrase*[3]. This shift is based on the observation that, in some languages, "elements that can serve as the nucleus of a referring expression are categorially varied" (Van Valin 2005, 28). In more common terms, it means that NPs are observed to have non-nominal elements as their heads. Examples cited in Van Valin (2005, 28) come from Nootka (Wakashan, Canada), but other incorporating languages may exhibit the same degree of part-of-speech fluidity. However, since Van Valin (2005) uses the label NP throughout the book, and since most of the arguments in most of the languages can be regarded as headed by some sort of nominals, the term RP has not gained standard acceptance. Namely, the notational tradition by Bladier et al. (2022) (which I follow) continues to use the term NP.

[3] In Van Valin (2005), the term *referential phrase* was used. Since Van Valin (2008, 168), the term *reference phrase* has replaced it.

2.4.4 The operator projection

In RRG, operators are all grammatical categories that modify a sentence. They have different scopes, i.e., modify different sub-parts of a sentence. Therefore, they attach to the level over which they exert their scope. Similarly, nominal operators scope over different parts of the NP and thus attach to the different levels of the phrase.

In classical RRG, the so-called *operator projection* is a separate tree that does not interfere with the *constituent projection* discussed above (i.e., NUC, CORE, etc.). In newer annotation practices, operators are integrated into the main tree using a special 'OP' tag.

Operators are not the focus of the current work, so I will refrain from providing further information on them for brevity.

2.4.5 The layered structure in head-marking languages

Head-marking languages also have a layered structure[4]. The comprehensive analysis of head-marking languages is presented by Van Valin (2013). The key point in analyzing head-marking languages (and double-marking languages having some head-marking properties) is recognizing bound verbal morphemes as actual core arguments (in contrast to the NPs, which are core arguments in dependent-marking languages). The reason is that in head-marking arguments, the verbal complex constitutes a correct, complete sentence, see (1).

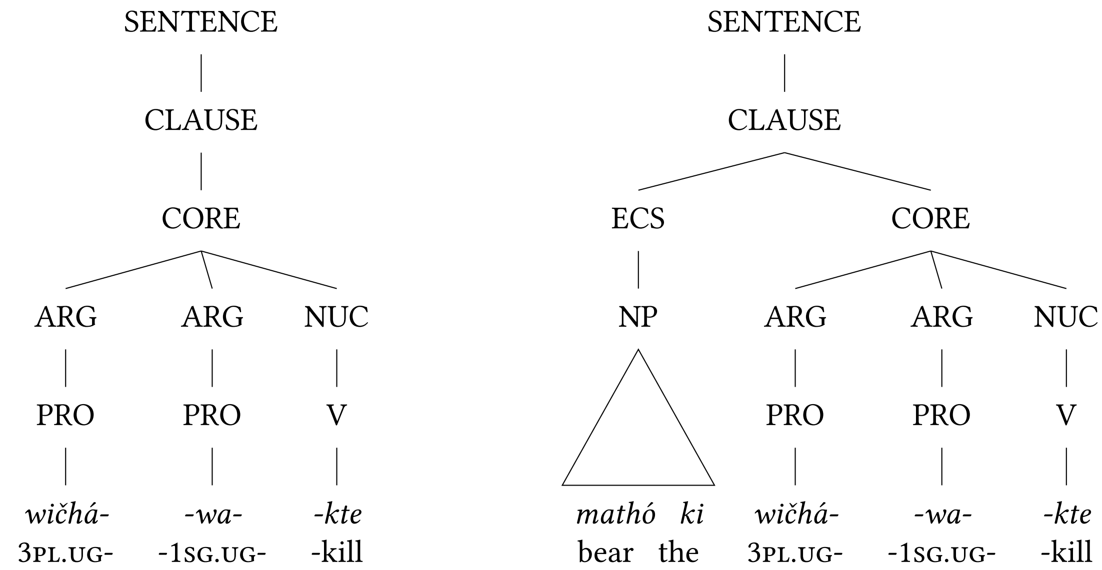
(1) Wičhá-wa-kte — LAKHOTA
3PL.UG-1SG.A-kill
'I killed them.' — Van Valin 2005, p. 147, (5.9a)

Example (1) is a complete sentence in Lakhota (Siouan). Therefore, it includes all the necessary layers, meaning morphemes are considered syntactic constituents in the absence of separate words. Consequently, in (1), the verbal root is the nucleus, and the prefixes are CORE arguments. The tree representation of (1) is given in Fig. 4a.

The status of the NPs in head-marking languages requires further discussion. For example, (1) can be augmented with an NP specifying the undergoer (the killed); see (2). In this case, the sentence (2) appears to have two constituents denoting the undergoer: the NP *mathó ki* 'the bear' and the prefix *wičhá-* '3PL.UG'.

(2) Mathó ki wičhá-wa-kte — LAKHOTA
bear the 3PL.UG-1SG.A-kill
'I killed the bears.' — Van Valin 2005, p. 147, (5.9b)

[4]This section briefly mentions my talk about head-marking and dependent-marking properties in Halkomelem (Generalova 2019a). Although it is not my only work on Salish languages (Generalova 2020c), due to a shift in the thesis's focus since 2020, I do not include a dedicated section on Halkomelem or other Salish languages.



a. The layered structure of (1)

b. The layered structure of (2)

Fig. 4. The layered structure of Lakhota sentences (1) and (2) (reproduced from Van Valin 2005, 147, 148 with modifications)

While the entity being killed is undeniably a core argument of the verb "kill," the NP in (2) is optional, functioning as if it were an adjunct. Van Valin (2013) suggests defining a special position outside the CORE and within the CLAUSE for argument-specifying NPs in head-marking languages, named Extra-Core Slots (ECSs). The CLAUSE-internal position of the ECSs allows operators to scope over them. Classifying them as external to the core resolves the issue of argument doubling: the core contains a unique set of arguments realized as verbal morphemes, while any duplicates reside outside the core. The tree representation of (2) featuring this approach is offered in Fig. 4b.

It is important to distinguish Extra-Core Slots from Pre/Post Core Slots. Van Valin (2013, 104) summarizes six key differences between them. First, Pre/Post Core Slots embed elements with a special discourse-pragmatic function (e.g., topicalized), while the functions of ECSs are not restricted. Second, entailed by the first, there can be only one Pre/Post Core Slot per clause since only one constituent usually has the special discourse function. In contrast, the number of ECSs is not restricted to one and is limited by the number of verbal arguments. Third, Pre/Post Core Slots are positionally restricted and usually are clause-initial (or right after the *detached position*). In contrast, ECSs can appear anywhere in the clause. The fourth important difference is that, if topicalized, adjuncts can occupy the Pre/Post Core Slot. ECSs are reserved for argument specification only. Adjuncts in head-marking languages are analyzed in the same way as in dependent-marking languages. Finally, Pre/Post Core Slots are found in both head-marking and dependent-marking languages, whereas ECSs are restricted to head-marking languages only.

The paper Van Valin (2013) received criticism from Martin Haspelmath (Haspelmath 2014). Haspelmath points out that Van Valin's analysis applies only to prototypical head-marking languages. Indeed, this analysis does not account for double-marking languages or constructions, which involve the co-occurrence of markers on both the head and the dependent within the same sentence[5]. According to Haspelmath (2014), the most complex situation arises with subject agreement in many dependent-marking languages, which should, from a comparative perspective, be regarded as a form of head-marking. As for RRG, it suggests that, in this case, the reasons for simplicity take precedence over those for comparability. As a result, RRG does not apply head-marking reasoning to obvious cases of verbal agreement, especially in languages that lack other examples of head-marking. However, this approach obscures the analysis of less straightforward cases; see discussion of Kikuyu (also known as Gikuyu, ISO 639-3 code: *kik*; Bantu, E51 in Maho (2009)) and Kinyarwanda (Bantu, JD61 in Maho (2009)) in Section 6.2.1.3. Therefore, I concur with Haspelmath (2014) that the existing RRG analysis of head-marking languages is incomplete. However, I do not see this incompleteness as a significant obstacle to using and implementing this analysis. In my thesis, I engage with the debate on head-marking from both theoretical and computational perspectives, offering observations and analyses that may benefit the development of a more comprehensive theory.

2.5 Arguments

2.5.1 Direct vs. oblique arguments vs. adjuncts

The basic layered structure of the clause offers a dichotomy between core arguments and (peripheral) adjuncts. However, some languages feature elements that are intermediate between these two categories.

In RRG, a distinction is made between *direct core arguments* and *oblique core arguments*, based on their coding.

> Direct core arguments are morphologically unmarked or coded with a direct case, as in dependent-marking languages like English and German, or are cross-referenced on the verb, as in head-marking languages like Lakhota and Tzotzil. Oblique core arguments are those marked by an adposition or an oblique case. Van Valin (2005, 57)

The theory already defines the list of cases: NOM, ACC, ABS, ERG, and DAT are classified as direct cases, while everything else is considered oblique. However, this definition does

[5]Aiming to expand Van Valin's analysis to less prototypical languages, I have examined this issue in Halkomelem (Salish) (Generalova 2019a), but this essay is far from a comprehensive RRG analysis of double-marking phenomena/languages.

not appear comparative (in Haspelmath's terms, see Haspelmath (2010)). For instance, different glossing traditions and the diversity of language structures can lead to the broad application of the DAT label to a wide range of NPs. Would all of these be classified as core arguments? Moreover, do languages lacking DAT have fewer direct cases, or is the role of the DAT taken by another case? Considering these limitations, defining the distinction between direct and oblique core arguments solely on the basis of "oblique" coding raises some issues.

However, the distinction between direct and oblique core arguments has important syntactic implications. Only direct core arguments are eligible for macrorole assignment (see below). This means that the status of an argument is closely linked to its macrorole status. The relationship is bidirectional: if there is evidence that an argument is assigned a macrorole, it must be a direct core argument. Conversely, if no macrorole is assigned, it may be either a non-macrorole direct argument or an oblique core argument.

A genuine intermediary level between arguments and adjuncts is one of the so-called "argument-adjuncts". Initially, they were postulated for prepositional phrases that "contribute an important component of the meaning of the sentence" (Van Valin 2005, 23). Consider several sentences with the verb *put*: *Max put the book down / there / on the table / behind the TV*. The location where the book is put is semantically required by the verb and thus resembles an argument. On the other hand, it is not constrained semantically by the predicate and thus resembles an adjunct. In the newer annotation tradition (Bladier et al. 2022), such elements appear in the main tree representation of a sentence but are not tagged as PERI (unlike true adjuncts).

In the present work, I begin my analysis with the assumption that constructions with morphological causatives based on transitive verbs have three direct core arguments. This hypothesis is grounded in the principle of simplicity: the simplest explanation is usually correct unless the opposite is proven. Some exceptions to this assumption were identified in my language sample and are discussed in the respective sections of Chapter 6.

2.5.2 Macroroles

The concept of macrorole is an innovation introduced by RRG. There are two macroroles postulated: *actor* (A) and *undergoer* (UG). Essentially, these macroroles serve as generalized labels for the most agent-like and patient-like participants, respectively. According to Van Valin (2005, 60), macroroles "correspond to what is pretheoretically commonly called 'logical subject' and 'logical object' ".

The motivation behind introducing macroroles is the idea that the similarities between semantic constructions stem from more abstract notions than semantic (thematic) roles; consider the example below. Additionally, syntactic effects, such as topicalization or verbal derivation, can also be generalized in terms of these two macroroles.

> Agent, effector, experiencer, perceiver, possessor, judger, etc., can be the subject of an active verb, while patient, theme, stimulus, possessed, location, etc., can be direct object. In English, passive, patient, theme, stimulus, possessed, location, etc., can be subject, while agent, effector, experiencer, perceiver, possessor, judger, etc., can be the object of the preposition by. It appears that a significant generalization is being missed here since there are long disjunctive lists of roles in these statements. But in fact, it is not an accident that they seem to group together the way they do, and the obvious generalization can be captured in terms of semantic macroroles: in an active clause, the actor is the subject and the undergoer direct object, while in a passive the undergoer is subject and the actor is in a peripheral PP. Van Valin (2005, 60–61)

The principles of macrorole assignment dictate both the number of macroroles a verb can take and their nature. It is generally assumed that verbs with two or more arguments take on two macroroles. In contrast, verbs with a single argument assume one macrorole—actor for activity predicates and undergoer for non-activity predicates. Nevertheless, there are numerous cases of discrepancies. Certain languages, and even individual verbs, can have specific rules governing macrorole assignment. Throughout this thesis, I will reassess the assignment of macroroles in the languages included in my sample.

ACTOR	$\longrightarrow$		$\longleftarrow$	Undergoer
arg. of **DO**	1st arg. of **do′** (x,...)	1st arg. of **pred′**(x, y)	2nd arg. of **pred′**(x, y)	arg. of state **pred′**(x)

Table 2: The Actor-Undergoer Hierarchy (reproduced from Van Valin 2023b, p. 108, (99))

To account for the likelihood of a given argument becoming an Actor or Undergoer, RRG postulates an Actor-Undergoer Hierarchy, see Tab. 2. It is a bidirectional scale where all semantic arguments are located by the prominence of their Actor-like and Undergoer-like properties. These properties are described in terms of RRG's logical structures but bear a resemblance to the proto-properties of a proto-agent and proto-patient as suggested by Dowty (1991). The arrows on the scheme indicate the "increasing markedness of realization of argument as macrorole". Thus, the arguments of **DO** (which stands for an activity predicate) are most prototypically realized as Actors, while arguments of states are usually realized as Undergoers. The realization of the arguments positioned closer to the center of the scale is more variable.

I find the bi-directionality of this scale one of the most significant and valuable innovations. Usually, hierarchies are formulated so that they have one single direction. Once a trend at the "lower" end is discovered, the hierarchy is claimed to operate "vice versa". The most illustrative example would be case assignment in ergative (or split ergative) languages that select not the highest but the lowest argument as the subject and the second-lowest (or second-highest?) argument as the direct object. However, the "vice versa" logic is not always as straightforward as it might seem. Therefore, unidirectional hierarchies sometimes fail to predict systematic linguistic processes taking place at their "lower" end.

In contrast, the Actor-Undergoer Hierarchy in RRG offers a more versatile approach to addressing each argument. The arguments at the extremes of the scale are intended to be equally prototypical, meaning they are equally eligible for a particular syntactic effect. The choice of which end of the scale serves as the starting point depends on the language in question. Therefore, prototypical nominative-accusative languages count from left to right, prototypical ergative-absolutive languages count from right to left, and some languages (such as active languages) count simultaneously from both ends toward the center.

The Actor-Undergoer Hierarchy, as depicted in Tab. 2, highlights only the most prominent cross-linguistic generalizations. However, individual languages, constructions, and even specific lexemes can display unique patterns ("idiosyncrasies") when assigning macroroles. Part of that variety is studied in the present work.

There is an ongoing debate about whether a third macrorole should be postulated. Some believe it must be used for a second object of a prototypical ditransitive construction. Others suggest that it must be used for recipients. Although these two concepts are usually linked in European languages, this connection does not hold for many other languages worldwide. Moreover, languages demonstrate considerable variability in how these roles are assigned and coded. Both these suggestions are against the spirit of the very notion of a macrorole, which has been introduced as independent of grammatical relations and semantic roles. Both suggestions challenge the fundamental idea of macroroles, which was introduced to exist independently of grammatical relations and semantic roles. Macroroles do not map onto grammatical relations; rather, grammatical relations reflect a language's decisions regarding the encoding of macroroles.

Van Valin (2004, 79–81), who addresses the number of macroroles explicitly in the last section, argues that there are three main obstacles to postulating the third macrorole. First, "it would not be universal" since many languages do not allow three-argument cores at all, and other languages do not have a dedicated "indirect object" position. In contrast, actors and undergoers are present in all languages. Second, "it would not receive consistent morphosyntactic treatment". While actors and undergoers indeed vary across languages, a consistent trend exists: they are never oblique in a simple active clause. No such pattern is observed for recipients, instruments, or alternative candidates proposed for a third macrorole. The final argument presented by Van Valin (2004, 81) is that the third macrorole "would be relatively unimportant for the syntax" meaning that this notion is irrelevant for "the subject selection of intransitive verbs" and for "the major typology of syntactic systems". I find this argument the weakest of all because it is not required that all the notions in a theory are relevant for the same set of phenomena. However, I see the point that if macroroles are designed for "the major typology" or a coarse-grained classification of languages, notions that do not contribute to that should be separated to avoid complicating the analysis. Nevertheless, the typology of ditransitive, instrumental, causative, or any other constructions leads to the development of "the major typology". It would require its apparatus, even if it is not as universally applicable as macroroles.

This thesis primarily focuses on three-argument constructions and the assignment of macroroles within them. Even among a small sample of languages, there are significant differences in how the three arguments are treated. Perhaps, this work becomes one more argument against the necessity of a third macrorole.

2.5.3 Case assignment

Since case assignment is critical in my work, I have decided to collect the most important claims RRG makes about it in a separate section.

RRG collects generalizations and formulates them as *rules* that hold cross-linguistically in most prototypical constructions. Less prototypical language-specific or predicate-specific effects are called "idiosyncrasies" and described as modifications of these rules.

Concerning simple active clauses, the case assignment for their participants depends on the basic alignment of the construction in question. At this point, RRG carefully refers to accusative (resp. ergative) *constructions* rather than languages since the rules apply to individual sentences rather than entire syntactic systems. Therefore, regardless of the language's general trend, each construction receives proper treatment. The general rules for prototypical accusative (see (3)) and ergative (see (4)) simple active M-transitive clauses are given below. Note that the (b) claim of each of these rules does not apply when there is no second macrorole.

(3) Case assignment rules for accusative constructions:

a. Assign nominative case to the highest ranking macrorole argument (in terms of Tab. 2).

b. Assign accusative case to the other macrorole argument.

Van Valin (2005, p. 108, (4.26))

(4) Case assignment rules for ergative constructions:

a. Assign absolutive case to the lowest ranking macrorole argument (in terms of Tab. 2).

b. Assign ergative case to the other macrorole argument.

Van Valin (2005, p. 108, (4.26))

Non-prototypical case assignment (or "quirky case", e.g., dative subjects) is also handled by RRG using macrorole-based reasoning. I refer to it in my work when discussing Bantu causatives (Section 6.2.1.2).

RRG also provides rules for the prototypical assignment of dative and instrumental cases; see (5).

(5) a. Assign instrumental case to non-MR *b* argument if, given two arguments, *a* and *b*, in a logical structure, with (1) both as possible candidates for a particular macrorole and (2) *a* is equal or higher (to the left of *b*) on the AUH, *b* is not selected as that macrorole.

b. Assign dative case to non-macrorole arguments (default).

Van Valin (2005, p. 110, (4.29))

The reason for formulating these two rules together is that both dative and instrumental phrases occur as a third, non-macrorole core argument in many languages, whether accusative or ergative. Moreover, Van Valin (2005, 110) notices that the instrumental case "has specific conditions on its application", while the usage of the dative is much broader. Therefore, he formulates special conditions for using INS in (5) and, in the (b) claim, suggests using DAT as a *default* case. The term "default" might seem misleading in this particular formulation. I would recommend using the term "elsewhere" instead. Thus, all scenarios where there are three core arguments and the instrumental rule fails to apply should be considered "elsewhere" situations, necessitating the use of the dative case.

The rule for the instrumental assignment might be complex, hence the example (6). The logical structure in (6b) includes two effectors (arguments of the structure [**do′** (x, ∅)]), namely, *žena-* 'woman' and *ključ-* 'key'. They are positioned at the same rank on the Actor-Undergoer Hierarchy and thus are equally eligible for a given macrorole, namely, the actor. However, there can be only one actor in a simple clause. Therefore, the first of the two candidates – *žena-* 'woman' – is chosen as Actor and receives NOM, while the second candidate remains the non-macrorole core argument. This scenario aligns precisely with the situation described in (6a), which predicts the assignment of INS case.

(6) a. Žena-∅ je otključa-l-a vrata ključ-em. CROATIAN
woman-F.SG.NOM be.3SG unlock-PST-F door.PL.ACC key-M.SG.INS
'The woman unlocked the door with the key.'

b. [**do′** (*žena-*, ∅)] CAUSE [[**do′** (*ključ-*, ∅)] CAUSE [BECOME NOT **locked′** (*vrat-*)]] Van Valin (2005, p. 110, (4.30))

The assignment of the dative as a "default" case has been debated a lot (I believe the misleading term contributed to that). It is important to clarify that the "elsewhere" assignment option is not the only way the dative case appears in a language. Some specific constructions, predicates, or prepositions may require the dative case regardless of context and thus do not fit the situation described in (5). The "elsewhere" clause is called only in a simple active clause with three core arguments, one of which naturally bears no macrorole.

An interesting approach towards assigning instrumental and dative cases is presented in Nakamura (2021). The author suggests differentiating **direct** non-macrorole arguments from other "case-bearing elements other than macrorole arguments" in (7) and resolving the conflict "in terms of Jakobson's featural definitions" (Nakamura 2021, 135, Jakobson 1936).

(7) a. All case-bearing elements other than macrorole arguments receive instrumental case.

b. Non-macrorole direct core arguments that do not involve a violation of the AUH receive dative case. Nakamura 2021, p. 163, (14)

To avoid confusion and information overload, I will not review this approach in this introductory chapter. However, I will refer to it and to other works by Wataru Nakamura throughout my study.

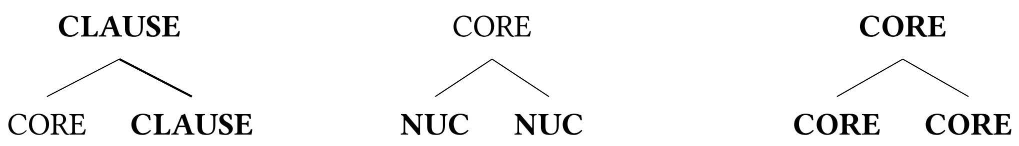
2.5.4 Closing remarks on arguments in RRG

There are thus three levels in RRG that are essential for analyzing the argument structure of a predicate. First, each predicate has a *semantic valence*, which indicates the number of semantic arguments it can take. Semantic arguments remain constant in all syntactic realizations of a predicate. In other words, semantic valence is a lexical feature, i.e., a characteristic of the verb as a lexeme.

Second, there is the syntactic valence, which denotes the number of direct core arguments. RRG uses the term *S-transitivity* for referring to this level. The S-transitivity of a predicate can vary depending on the type of construction being used and the nature of the arguments involved. Specifically, the S-transitivity of a verb is reduced by one when one of the arguments is expressed as a complement (instead of an NP), cf. (8) from Van Valin (2005, p. 255, (7.43b)) (following Foley and Van Valin (1984)), modified.

(8) a. Beckie wants a new Porsche. → 2 core arguments

b. Beckie wants to buy a new Porsche. → 1 core argument



a. Clausal subordination b. Nuclear coordination c. Core cosubordination

Fig. 5. Examples of nexus-juncture types

Third, there is the so-called *M-transitivity*, which tells how many macroroles a predicate takes. This number cannot exceed two, as only two macroroles are available. Consequently, there will always be a discrepancy for predicates with three semantic and syntactic arguments. Furthermore, even for predicates with one or two semantic and/or syntactic arguments, the number of macroroles is not necessarily the same. While I will not explore all instances of these discrepancies in detail, I will pay special attention to the three aspects of valency and transitivity outlined in RRG throughout the following chapters.

2.6 Juncture-nexus types

2.6.1 General presentation

There is a detailed approach to non-simple predications within RRG. All complex structures, i.e., structures with multiple predicates, are described in terms of *nexus* and *juncture*. The term **juncture** refers to the layer of the clause that appears multiple times in a sentence. Depending on the number of shared arguments and operators, structures that are commonly referred to as 'complex clauses' or 'biclausal sentences' fall into **bi-nuclear**, **bi-coral**, **bi-clausal**, and **bi-sentential** events. The term *nexus* relates to the type of relationship between the two repeated layers. RRG distinguishes three relations: *coordination*, *subordination*, and *cosubordination*[6]. One of the distinctive features of RRG is that these relationships are conceptualized independently of their structural realization. For example, the coordination relationship does not necessarily imply a syntactic conjunction. Sample trees for the three relations are shown in Fig. 5

The *subordinate* element is thus defined as embedded into the logical structure of the matrix predicate. In contrast, neither coordination nor cosubordination contains any embedded elements. Cosubordination requires obligatory sharing of operators. Therefore, if an operator scopes over two units at the same level, the relationship is tighter than that of coordinate structures. Thus, there is a tightness scale: cosubordination → subordination → coordination. The second dimension of this scale is provided by the

[6]The *cosubordination* relation has been criticized by Foley (2010) and Bickel (2010), see also a reply to them in Van Valin (2021).

	Cosubordination	Subordination	Coordination
Nucleus	Causative [1] Mary wiped the table clean.	Causative [2] Mary made Sam wash the dishes.	Endeavour Fred tried to make a cake without a mix.
Core	Psych-action Mary decided to take the job offer.	Propositional attitude Bill wants very much for Sam to visit Nancy tomorrow.	Causative [3] Maria taught Fred to speak Portuguese.
Clause	Intentionality Max considers Don to be a loser.	Speech Frank said that his friends are corrupt.	Location/Time Sally met Bob for a drink after she left the office.
Sentence	(does not exist)	Circumstance The baby cried, because it was hungry.	Temporality Lisa yawned loudly while Dan was talking.

Table 3: Semantic relations conveyed by all nexus-juncture types with English examples (modified from Van Valin 2023b)

structural layers: nucleus → core → clause → sentence.

Each combination of nexus and juncture is claimed to express given semantic relations. These are shown and exemplified with English sentences in Tab. 3. Although the general trend from the strongest to the weakest link between the two components may hold across languages, the placement of each relation may not be the same as it is in English. Notable findings about complex structures emerged from the work on the RRG parallel treebank (Bladier et al. 2022). While the list of relations in Tab. 3 is not exhaustive (see Van Valin (2023b) for more), it serves an illustrative purpose here.

2.6.2 Causative constructions as nexus juncture types in classical RRG

1. Causative [1]: one state of affairs makes another state of affairs happen, and the two are subparts of a single state of affairs.
2. Causative [2]: one state of affairs causes another state of affairs to happen, and the causing state of affairs is distinct but unspecified from the caused state of affairs.
   (a) Unmediated: *Mary made Sam wash the dishes.*
   (b) Voluntary: *Mary had Sam wash the dishes.*
   (c) Facilitating: *Mary let Sam wash the dishes.*
   (d) Assisting: *Mary helped Sam wash the dishes.*
3. Bringing about
   (a) Causative [3]: the bringing about of a state of affairs by a distinct state of affairs.
   (b) Jussive: a verbal causative, i.e., the expression of a command, request or demand.
   (c) Permissive: making a state of affairs possible by removing a barrier to it or directly licensing it.
   (d) Injunctive: making a state of affairs impossible by putting up a barrier to it or directly enjoining it.

Table 4: Types of causatives (according to Van Valin 2023b, 84–86)

What is crucial for the topic of the present work is that this table presents three distinct types of causatives, which are predicted to correspond to various kinds of complex structures. The complete definitions of these types are provided in Tab. 4. As illustrated in Tab. 4, the causatives are distinguished by the specificity of their causative meaning. In Causative[1], the causative meaning is very general. The subtypes of Causative[2] offer some details on how the causer acts, but it nevertheless is physically involved in the process. The type referred to as "bringing about" contains even less prototypical causative constructions, in which causation requires some additional entities. The difference between these types of causatives is often reflected in language structures: Causatives[1] tend to be morphological, while Causatives[3] are usually analytical.

However, I will analyze constructions featuring a morphological causative as simple clauses rather than junctures throughout the present work. One reason for this choice is that I focus on morphological causatives that are expected to exhibit the tightest possible link. Interestingly, the tightest link can also imply its absence. The second reason is the established mono-eventive nature of the causatives in the languages included in my sample. It has been noted that some morphological causatives (e.g., Japanese *-(s)ase* causatives) function more like syntactic causatives as if constructed with the aid of two independent verbs. These constructions are sometimes referred to as *biclausal* or *bieventive*. Although I occasionally use these terms, particularly when citing other works, they are not entirely relevant in RRG. Thus, they can be effectively replaced by a simple clause or a nexus-juncture type.

2.7 Constructional schemas

The previous sections discussed cross-linguistic generalizations that form part of the theoretical analysis, along with some deviations that are specific to certain languages. To explain these "language-specific representations of morphosyntactic information" (Van Valin 2023b, 125), RRG employs a special device called *constructional schemas*. These schemas are not fully formalized data structures designed to describe how a specific construction deviates from the standard expected by the basic analysis across various linguistic dimensions—be it morphology, syntax, semantics, or pragmatics. In particular, they can encapsulate all non-canonical macrorole assignment and case selection rules. The types of constructions that can be described using constructional schemas are quite diverse. For instance, Van Valin (2005, 132–135) proposes schemas for passive, antipassive, reflexive, and WH-questions in English and Sama.

However, it is important to note that classical RRG does not differentiate between *language*-specific and *construction*-specific phenomena. For example, Icelandic (Van Valin 2005, 119) is said to exhibit a non-standard passive construction, with the linking in the passive construction expressed as *~A = PSA* rather than the expected *U = PSA*. Should this distinction be captured in the constructional schema for the Icelandic passive, or should it reflect characteristics of the Icelandic language as a whole? A comprehensive theory that aims to be cross-linguistically comparative needs to have such questions clarified.

One could object that "idiosyncratic, language-specific features of constructions" Van Valin (2005, 132) need not and can not be comparable. Nevertheless, the criteria for determining what constitutes "idiosyncratic" remain somewhat ambiguous. It hinges on how much a construction deviates from standard rules. Having a standardized schema for a typical construction could facilitate the identification of such deviations.

A similar point is made by Diedrichsen (2011, 183), who observes that while constructional schemas exist for "passives, antipassives, conjunction reduction, and wh-questions," there are none for intransitive, transitive, or ditransitive constructions. To enhance the constructional dimension within RRG, a standardized constructional schema that accounts for both standard and idiosyncratic realizations of each construction in a comparable way seems essential.

> My suggestion is to use the constructional schema as a general descriptive tool for constructions. It contains explicit information about the fixed structural elements that would help the speaker/hearer recover the construction in the online production of utterances. As speakers also know about the productivity and variability of constructions by convention, the constructional schema has to inform about variable elements of the construction as well.
>
> The advantages of such an extension of the theoretical impact of constructional schemas include new compatibility of RRG for the description of syntactic change and variation. Thus, this approach facilitates the description of grammaticalization phenomena in RRG. The entire issue of spontaneity, change, novelty, and variation, which has been missing in previous approaches to RRG, can be fitted easily into the theoretical framework without changing its major properties. Diedrichsen (2011, 183–184)

However, Diedrichsen and Nolan, in their works, advocate for a rather radical constructionist transformation of RRG (Van Valin and Diedrichsen 2006; Diedrichsen 2012; Nolan 2011; Nolan and Diedrichsen 2013). Specifically, they propose abandoning the concepts of macroroles and case assignment rules, opting instead for a comprehensive listing approach to describing a construction.

> The Macrorole assignment principles, the Case assignment rules, the PSA selection principle and verbal agreement can be neglected, as all of this is ruled by the construction. There would, however, have to be some statement about the cases that are distinguished in a language and the distribution of overt vs. non-overt case marking. Diedrichsen (2011, 186)

In this context, I want to highlight the difference between a linguistic theory and its applications. While macroroles and other rules may seem unnecessary in some computational implementations of RRG (Nolan and Salem 2010), they remain a powerful theoretical device for explaining various language phenomena. If one wants to build an RRG-inspired constructicon (a "lexicon" of constructions) of a language, one might prefer simply listing examples in a schema. In contrast, if the goal is to introduce more constructionist reasoning into the theory of RRG, it is essential to reinvent constructional schemas in ways that align with other theoretical findings.

In Chapter 10, I apply constructionist reasoning to account for the different argument-marking strategies in the languages within my sample, interpreting them as distinct causative constructions, each with its own morphosyntactic characteristics. Nevertheless, all these constructs represent specific instantiations of an abstract generalization created using other RRG mechanisms.

2.8 The linking algorithm

The Linking Algorithm is considered "central to a theory like RRG" (Van Valin 2005, 128), accounting for both canonical and non-canonical phenomena. Generally, it links syntax and semantics. The term "algorithm" reflects the nature of this system: in Van Valin (2005), it is formulated as a set of imperative instructions (like an algorithm in a command programming language): *determine the logical structure, assign a macrorole, select a syntactic position…* Importantly, RRG claims to use the same algorithm in both directions: syntax-to-semantics and semantics-to-syntax.

> Viewed in terms of a language processing model, the semantics-to-syntax linking is an aspect of the production process, while the syntax-to-semantics linking is an aspect of the comprehension process. Van Valin (2005, 129)

However, there are nuances specific to each direction, and the syntax-to-semantics algorithm presented in Van Valin (2005, (5.10), 149–150) is not a simple mirror image of the semantics-to-syntax algorithm Van Valin (2005, (5.5), 136).

The linking algorithm provides a linear order of using all the RRG devices mentioned in this section (and additional ones that are not of primary importance for the present thesis), such as the Actor-Undergoer Hierarchy, case assignment rules, and constructional schemas. However, this order does not truly reflect how sentences are produced or processed in human communication, and it was never intended to do so.[7] Thus, the procedural nature of the classical linking algorithm does not appear to be its most essential quality, suggesting it need not be retained in future work on linking. For instance,

[7]For more details on how classical RRG can be modified to serve as a basis of a processing model, see Van Valin (2006) and Van Valin (2023a).

Theory		Sentence analysis
Logical structures	$\longrightarrow$	semantic representation
$\updownarrow$	AUH $\cap$ ConstrSch	
Macroroles	$\longrightarrow$	MR status of each argument
$\updownarrow$	Case rules $\cap$ ConstrSch	
Syntactic trees	$\longrightarrow$	tree representation

Table 5: Modules of RRG and their outputs

Osswald and Kallmeyer (2018) proposed a formalization of RRG based on static models and a set of constraints applied to particular components. I discuss this formalization in Chapter 8 and develop my implementation based on it in Chapter 10.

Since I do not use the classical linking algorithm in my work, I will not present it in detail here. However, I find it necessary to summarize the RRG principles and devices relevant to linking and both parts of the present thesis (theoretical analysis and computational implementation), see Tab. 5. This graph is based on Van Valin (2005, Fig. 5.1., 129) and the accompanying description, but is modified to clearly separate the theoretical devices of RRG (on the left-hand side) and the types of sentence analyses they produce (on the right-hand side). Thus, to analyze a sentence with RRG, one needs to create three representations: a semantic representation, a syntactic (tree) representation, and a macrorole status for each argument (the operator projection and the information structure are not considered in Tab. 5). To generate these representations, RRG has three theoretical modules that describe logical structures, macroroles, and syntactic structures, respectively. As stated in Van Valin (2005, 128), "the relation between logical structure and macroroles is mediated by the actor–undergoer hierarchy". In Tab. 5, the actor-undergoer hierarchy (AUH) is accompanied by a constructional schema (ConstrSch). This means that, from a theoretical perspective, the additional requirements listed in constructional schemas can override some of the standard principles listed in the AUH. When analyzing individual sentences, it is essential to determine whether each sentence follows the canonical trend or is a specific construction that requires deviations from the canonical pattern. The same mediating structure exists between the macrorole module and the syntactic module. These are default rules of case assignment, but already Van Valin (2005, 128) admits that this part "is subject to extensive cross-linguistic variation". Thus, the constructional schemas that establish additional constraints for this level are more numerous, and their contribution to the analysis of individual sentences is correspondingly more significant.

Chapter 3

Prior approaches to (three-argument morphological) causatives

3.1 Introduction and motivation

The literature review is always an inevitable but usually not the most exciting part of every thesis. Nevertheless, the readers of the present work might find it more useful than usual. Since the present work is based on a specific theoretical framework (namely, RRG), it might seem detached from the mainstream of causative studies. Moreover, understanding complex matters in RRG is challenging even after being introduced to its basic notions.

Additional assistance might come from the "translation" of non-RRG papers known to many readers into a theory-independent and/or more RRG-compatible reasoning and language. These translations might pave the way through the current study for readers familiar with other theoretical backgrounds. Thus, this chapter serves as an entry point for readers: each presented source is summarised and related to RRG and the present study.

The primary focus of this thesis is on strategies of argument marking in three-argument constructions with morphological causatives. It determined the selection of works presented in this chapter. First, I do not review the early generative studies discussing the place of the CAUSE head in the structure of the predicate without discussing the arguments of such a predicate. Second, I do not review studies seeking to explain how the human mind conceptualizes causation, as they draw few conclusions relevant to processing sentences of a specific structure. Third, this chapter lists only studies aimed at making cross-linguistic generalizations (even though some are centered on one or two languages), not studies that explain variation within a single language variety or family. Such studies find their place in the respective sections of Chapter 6.

Causation, understood in a very broad sense – so that it also includes 'usual' transitive verbs – was one of the crucial topics in early generative studies. For the present work, this question is not relevant for several reasons. The main reason is that the theoretical framework of the present thesis (RRG) is very far from the generative approach and does not allow movement, deep layers, or implicit syntactic layers that play a significant role in the generative analyses of causatives. Nevertheless, I included some generative studies in Section 3.10 to showcase their reasoning. Moreover, some ideas from generative studies remain relevant to the current study, which I will aim to make visible in the relevant sections.

The chapter is structured as follows. It opens with a review of the first (to my knowledge) volume entitled "Typology of causative constructions" (Xolodovič 1969), which is not widely known among researchers and readers who do not read in Russian. Section 3.3 presents the study that has been highly influential in the Western world, Comrie (1976) There is an immense corpus of language-specific studies examining whether the "paradigm case" applies to their data or not. In the next section, I review two studies that explicitly challenge Comrie (1976) and still aim to advance it. Section 3.5 presents two studies that build on Dowty's theory of proto-properties (Dowty 1991). Since RRG is also grounded on Dowty's findings in many respects, these papers are especially close to it, and the transition is rather smooth. Section 3.6 accumulates various studies that discuss how the causee can be marked and the factors that lead to the selection of a given marker. These questions received far more attention than questions related to any other components of a causative construction. The papers within the section are sorted from historical and general to more recent and specific. Special attention is dedicated to the paper by Kozinsky and Polinsky (1993) that tackles a very particular question of identical coding (marking) of the causee and the affectee.

3.2 Xolodovič (1969): the first typological volume

One of the first systematic works about causative constructions is a collective volume by leading scholars of the so-called Leningrad Typology Group, edited by Xolodovič (1969). In this volume, a morphologically driven approach is adopted. The authors minutely describe how causative constructions are built in each language. The orientation of the work as a whole is more towards semantics. Except for just describing what is attested in a language, each author tends to give an exhaustive account of the range of meanings expressed by causative morphemes in a given language. Although they might seem outdated in some respects, studies in Xolodovič (1969) help narrow a study's object in typologically and theoretically salient ways.

The volume opens with two introductory articles. The first one (Nedyalkov and Sil'nitskiy 1969a) defines causative constructions and articulates theoretical and methodological concepts used in other articles of the volume. The description of causative con-

$$\underbrace{\underbrace{\overset{Your}{r_i}\qquad\overset{action}{s_i}}_{antecedent}\quad \overset{made}{k}\quad \underbrace{\overset{him}{r_j}\qquad\overset{go}{s_j}}_{consequent}}_{macro-situation\ (causative)}$$

Fig. 6. Causative situation (reproduced from Nedyalkov and Sil'nitskiy 1969a, 6)

structions in this paper is given in a formalized way. Many ideas are presented as numbered lists, tables, and formulae following a clear logic. As a result of this paper, a list of parameters of causative constructions that may vary across languages is provided.

According to Nedyalkov and Sil'nitskiy (1969a, 6), all causative constructions are representations of complex situations ("macro-situations") that are composed of two "micro-situations", united by the relation of causation. This general semantics of the causative situation is shown in Fig. 6, where *k* stands for the causative relation, *r* – for referents, and *s* – for states. The accompanying example shows that these components ("constants") are purely semantic and can be realized as various instances of a given language. For example, the relation of causation *k* can be expressed in a specific auxiliary verb, conjunction, morpheme on a verb, etc. Nedyalkov and Sil'nitskiy (1969a, 7) point out that any of these five basic constants may be omitted semantically or syntactically. The presence or absence of a "constant" in the semantic representation of a given construction does not entail its presence or absence in the syntactic one.

From the semantic point of view, Nedyalkov and Sil'nitskiy (1969a, 28) claim two "types of causation", each being formulated as a contrast. The first one is "factual" vs. "permissive". The negative variety of "permissive" causation is called "prohibitive". The core point of this opposition is the role of the causer. In "factual" contexts, it is the primary source of any changes, while in "permissive" contexts the causer lets (or not) them happen without much control.

The second type of causation is "contact" vs. "distant". As the authors explain, "in a situation of distant causation, there is an indirect connection between the causer and the caused state of affairs which conditions a greater or smaller autonomy of the causee in his acquisition of the caused state s_j". In a "contact" causative, no such autonomy is possible. An important note is that by its definition, the "permissive" causation can only be "distant". To sum up, these two binary oppositions lead to three different semantic types of causation, according to Nedyalkov and Sil'nitskiy (1969a): "permissive", "factual contact" and "factual distant".

Although semantic classification is not essential for my work, it is important for the overall reasoning about causative constructions. I report on similar claims in other works and compare them against these three types.

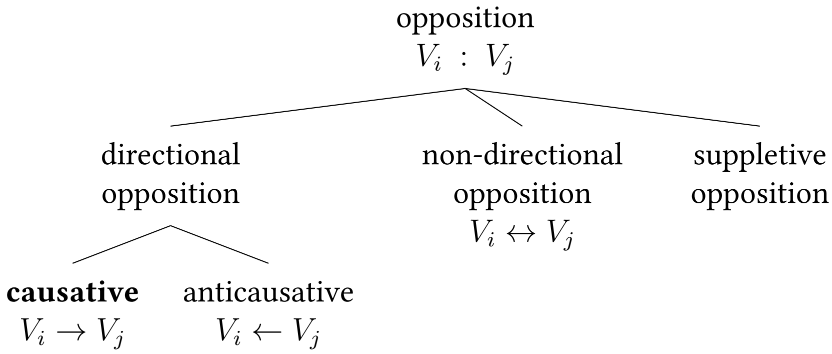


Fig. 7. Types of causative opposition (according to Nedyalkov and Sil'nitskiy 1969b, 20–22)

I find the volume by Xolodovič (1969) especially useful for defining and classifying morphological causatives. Therefore, I would like to present their approach in more detail.

The chapter dedicated to the typology of morphological and lexical causatives opens with the postulation of the following opposition, i.e., contrast.

> The causative opposition $V_i : V_j$, where V_i denotes the constant s_j (i. e, some kind of a state), and V_j denotes ks_j (the same state, but having been caused). Verbs V_i are non-causatives, and verbs V_j are their causatives. V_i and V_j are united with the relation of a semantic derivation: V_j is «derived» from V_i by attaching an additional meaning k.
>
> Nedyalkov and Sil'nitskiy (1969b, 20)

The opposition $V_i : V_j$ is very general and accounts for four large groups of situations briefly sketched in Fig. 7. The term *morphological* causative is used to denote the participant of the opposition "formally marked with a causative affix", i.e., the V_j in directional causative opposition (leftmost branch).

I follow this definition of morphological causatives in the present work. I also use the terms introduced above in the same sense.

Nedyalkov and Sil'nitskiy (1969b, 35–40) show that a causative morpheme either increases, leaves the same, or even decreases the syntactic valency of the base verb.

Cases in which syntactic valency decreases are pretty special. There are several cases of passive meaning introduced by the causative meaning and a complex matter of reciprocal causatives in Piro (Nedyalkov and Sil'nitskiy 1969b citing Matteson 1965, 39, 41, 82, 108). Without denying the importance of these phenomena for studies in causative constructions, I limit the scope of my study so that they are omitted.

The situation when the addition of a causative morpheme does not actually increase the syntactic valency of the verb is quite common. Even though I am primarily interested in modeling clear cases, i.e., when the syntactic valency increases following an increase in semantic valency, I also look at cases in which one of the participants appears as a peripheral argument or even an adjunct (without being included in the core). According to Nedyalkov and Sil'nitskiy (1969b, 50), this situation is persistent with verbs of feeling and verbs of transfer. The stimulus of a feeling verb takes a non-core position in some languages, but when it becomes a causer, it takes the syntactic slot of the PSA. However, Nedyalkov and Sil'nitskiy (1969b, 50) note that in these cases, it would be more accurate to talk about "the transformation of a non-obligatory syntactic valence to an obligatory" instead of postulating the maintenance of the base verb valency.

The situation in which syntactic valency increases alongside semantic valency seems very frequent across the world's languages. Nedyalkov and Sil'nitskiy (1969b, 47) suggest a table that captures all possible combinations of causative affixes with different bases (intransitive, transitive, and ditransitive). According to it, a three-argument construction can be built either upon a lexical transitive (types III and IV) or as a secondary causative construction from an intransitive construction (types I and II). I pay primary attention to derivations of type III and IV, where a distant or contact causative affix accompanies a lexically transitive base.

These basic notions and their formalized descriptions constitute the starting point of typological studies of causative constructions and the present study. Further review will elaborate on and refine these basics.

3.3 Comrie (1976): the most influential study

The volume "The Grammar of Causative Constructions" (Shibatani 1976a) was perhaps the first in the Western world to summarise the state of the art of causative studies at the time. Apart from the most cited paper by Comrie (see below), it contains several contributions divided into "Theoretical Foundations" and "Universals and Language-Particular Studies". Needless to say, in accordance with the reigning generative framework, theoretical foundations are grounded exclusively on English, making their "theoretical" findings barely useful for further studies. Papers in the second part are mainly descriptive, using the apparatus of the respective language-particular traditions. However, the first paper of Part Two is typological and has long been considered an indisputable standard of causative studies. With all due respect to these past studies and to the theoretical framework, which is very different from what I use, I would suggest that this volume be viewed more from a museum-like perspective. Both theoretical foundations and language-particular studies, not to mention universals, have advanced much over the decades, so these works play a little practical role, at least for the present work. Therefore, I offer a more detailed review of Comrie's paper alone, since it determined the central claim of many further studies.

The main discovery in Comrie (1976) is the so-called "paradigm case". According to Comrie (1976, 264), this generalization is mainly based on French and Turkish and aims to cover all languages, even though some contradictions are already noted in the paper itself. The "paradigm case" is characterized by four crucial properties. The first trait concerns the number of arguments in a causative construction. Comrie (1976, 264) claims that "no matter how many arguments a given noncausative verb has, there will also be an equivalent causative verb with one more argument". This appears not to be a valid generalization. Many languages (see Dixon and Aikhenvald (2000a) and Song (1996) *inter alia*) allow causatives from intransitive bases only. Since the present study is concerned only with three-argument constructions, I pay no further attention to this claim or its development in future research.

The second trait forbids the doubling of syntactic positions. The author attenuates this claim by claiming that doubling is allowed but is conditioned by a hierarchy: oblique arguments can be non-unique quite often, more restrictions apply to indirect and direct objects, and subjects are possibly never double (Comrie 1976, 276). One needs to delimit morphological, syntactic, and semantic reasoning to assess this claim critically. Thus, doubling of morphological devices (e.g., cases) does not necessarily entail the equality of the syntactic constituents (see Section 3.9 for details). However, the so-called *symmetrical* patterns (attested mainly in Bantu languages) exist and have two morphologically and syntactically identical constituents. I engage with this data in Section 6.2.1 and offer an alternative, RRG-based explanation for this evidence.

The third trait is important for formulating the distribution of various causative constructions in the world's languages. It predicts that the embedded subject (causee) is demoted if needed. The demotion of other participants is not predicted. Other typological observations (Dixon and Aikhenvald 2000a) and my investigation Chapter 6 tend to confirm this generalization. Citing K. E. Zimmer, Comrie (1976, 265) suggests that the demotion of the embedded subject affects the structure of the sentence least of all and is therefore cross-linguistically popular. I would like to support this statement by offering an explanation grounded on the RRG Actor-Undergoer Hierarchy (Van Valin 2005, 126). This hierarchy suggests that the closer a participant is to any end of the scale, the more properties of the actor (resp. undergoer) it has and, thus, the more stable it is syntactically. In contrast, participants located more towards the center of the hierarchy demonstrate high cross-linguistic diversity. In a three-argument causative construction, the causee is a less prototypical actor in comparison to the causer, so it is located further from the hierarchy ends. On the other hand, it is a less prototypical undergoer than the affectee. So, the causee is the closest to the center, i.e., the least prototypical participant, and thus, as predicted by the hierarchy, is the least stable in the construction.

The fourth claim is the notorious stepwise demotion. Comrie establishes four syntactic positions and orders them as follows:

(9) Subject → direct object → indirect object → other oblique constituents

The causee is predicted to occupy the highest (the leftmost) available slot. Thus, in causatives based on transitive verbs, the positions of the subject and the direct object are already occupied. Hence, the causee is predicted to become the indirect object. These notions are never discussed in Comrie (1976). The only remark is that "the syntactic positions DO, IO, etc. refer to the surface syntax of the sentences" (Comrie 1976, 263). The "surface syntax" apparently denotes a blend of morphological and syntactic properties (the RRG's initiative to discard traditional grammatical relations is motivated in large part by the desire to postulate more transparent concepts). Namely, the "indirect object" seems to correspond to a participant bearing the DAT case and eligible for some syntactic operations (to which participants bearing other cases are not eligible). However, this notion is arguable from a broader typological perspective. For some languages, the position of the indirect object is proven to be not pertinent (see, for instance, the debate in Say (2009) about Kalmyk). It means that no language evidence proves that an argument marked with a DAT (or similar case) is in any way different from arguments bearing other cases. It is then unclear what position a causee would take in this situation since all of them are equally free and equally available.

Moreover, there are languages where the causee does not demote to the indirect object position, even though it is free (and perhaps opposed to the position of an oblique object). Comrie (1976) calls this "extended demotion", opposing it to the stepwise demotion predicted by the paradigm case. These are multiple languages that allow causee to bear INS case or a different case/preposition used for agents of passive clauses. Such an example can be found in French (with an analytical causative).

(10) FRENCH

Je	fer-ai	mang-er	les	gâteaux	par	Jean
1SG.NOM	make-FUT.1SG	eat-INF	the.PL	cakes	by	John

'I will have John eat the cakes.'

Van Valin and LaPolla 1997, p. 533, (9.22b)

(11) FRENCH

La	maison	a	été	achet-é-e	par	des
DET.DEF.F.SG	house	AUX.3SG	be.PTCP.PST	buy-PTCP.PST-F	by	DET.PL

étranger-s.
foreigner-PL△

'The house was bought by the foreigners.'□

Larousse.fr. Dictionnaire de français Accessed: 2026-05-01

It must be highlighted that this marking vastly outnumbers all other cases among all "oblique" (i.e., not indirect) objects. Hence, saying that the causee demotes to any oblique position is a huge over-generalization. However, interpretations of this special position vary across studies. Comrie (1976) regards them as "agentive phrases" relating them to the realization of demoted agents of passive constructions. Mind the similarity (also reflected in translations) in (10)–(11). However, Comrie (1976, 271) draws a bizarre conclusion from this observation: "this agentive phrase is the result of passivization in the embedded sentence before causativization takes place". Today, it is clear that verbal derivations of passive and causative have no relation to each other. Hence, the fact that causees look like demoted agents in passive is conditioned by the semantics of these nominals or these cases (or the construction itself) but not by an additional step of verbal derivation.

Other papers regard INS-marked (and similar) causees differently. Namely, Kemmer and Verhagen (1994) relates them to instrumental clauses. In fact, it appears that languages vary with respect to whether they use different cases for prototypical instruments and demoted agents and whether any of those can mark causees (see discussion in Section 3.7).

It is important to highlight the question the paradigm case answers. In fact, the paradigm case tells what causatives derived from different types of base verbs should look like. It somehow presupposes a unique construction of each type: one possible causative-of-intransitive, one possible causative-of-transitive… However, there appears to be more than one option for each derivation type. In this light, the paradigm case appears to be a mere overgeneralization.

Having been highly influential, the paper Comrie (1976) has determined the vector of many further language-specific works: some tended to confirm the paradigm case, others sought to disprove it. These attempts sometimes obscured the description of the language data, and the present generation of scholars needs to uncover the actual data and reinterpret it.

3.4 Replies to Comrie (1976)

3.4.1 Song (1996) on NP density control

The book "Causatives and Causation" by Song (1996) is a sound typological study of causative constructions. It encompasses 408 languages, constituting a "convenience sample", rather than a representative one. The author justifies this decision with the aim "to characterize the formal variation of causative constructions, not to find correlations among different parameters. This aim is closely aligned with the present project's aim, even though my language sample is much smaller. However, the typology provided by Song (1996) is of secondary importance for the present study. In contrast, the idea of NP density control (presented in the same book and, earlier, in Song (1991)) proves to be quite helpful.

Song (1996, 135) argues against the traditional typology, which "is based crucially on the formal distance between [Vcause] and [Veffect]". This is a well-formulated reason for the distinction between the types *lexical* vs. *morphological* vs. *syntactical* causatives. The author criticizes this approach by pointing out that "the parameter of compactness does not contribute much to an understanding of how causation is cognized by the human mind" (Song 1996, 135).

In addition, Song offers a purely linguistic motivation for not regarding the compactness of a construction as its distinctive feature. The author notes that some degree of reduction is proper to any type of construction. Thus, compact causatives are not special types themselves but result from the grammaticalization of other types with different semantics (Song 1996, 134–135). This is, in fact, an excellent observation since grammaticalization is omnipresent in a multitude of language phenomena. Drawing attention to the fact that grammaticalization itself does not reflect any specific type of construction is important to detect the actual meaningful parameters of a given phenomenon.

Hence, Song (1996) suggests three types of causatives. The first type, called COMPACT, denotes constructions where "no other element can intervene between Vcause and Veffect" (Song 1996, 21). Namely, prototypical morphological and lexical causatives are claimed to belong to this type. On the other hand, Song (1996, 135–136) suggests two non-COMPACT types of causatives and claims them to differ in *implicativity*: "whether or not the truth of Seffect is built into the semantics of Vcause". In other words, in implicative causatives, the event denoted by the effect clause necessarily holds. With an implicative causative, it is impossible to run a "contradiction test", i. e., to negate the existence of the effect event. Song (1996, 136) cites the English sentence *John made Mary kick the man* as an example of an implicative causative and notes that the truth of its event clause *Mary kicked the man* holds whenever the truth of the whole sentence holds. The contradiction test predicts the unacceptability of **John made Mary kick the man, but Mary did not kick the man.* In contrast, with English make-causatives, such sentences are grammatically acceptable.

The same feature appears under the name of *defeasibility* in Copley and Wolff (2014) (see review below). In the present study, it proves to be relevant for explaining the argument structure of morphological causatives. I discuss this feature in detail in Section 6.1.1.3.

The first non-compact type of causatives suggested by Song (1996) – the AND-type – is formally built with two clauses having "an element coordinating the two clauses" (Song 1996, 35). Importantly, in all languages, the cause must precede the effect, not only linearly but also temporarily (Song 1996, 35). AND-causatives are claimed to be always implicative (Song 1996, 136). The second non-compact type of causatives suggested by Song (1996, 49) is PURP, which stands for purpose. The distinguishing property of this type is that the event of the Seffect "has not yet occurred"; the Seffect is a goal or purpose "to be realized by means of the event denoted by Scause". Nevertheless, Song

(1996, 136) claims that PURP causative can range in implicativeness from 0% to 100%. The COMPACT type is claimed to be a result of the grammaticalization of either of these types and, because of its compactness, is insensitive to implicativeness. Finally, Song (1996, 138) comes up with four prototypical types of causative constructions located at the edges of two scales: compactedness and implicativity. The author theoretically predicts the existence of various non-prototypical types of constructions, where one of the parameters (or both) is not totally present or totally absent, but, unfortunately, does not provide any quantitative support for the distribution depicted in Song (1996, Fig. 5.2, p. 138), see Tab. 6.

	non-implicative	implicative
compact	Kammu, Malagasy non-impl. morph. caus.	lexical / morphological causatives
non-compact	prototypical PURP causatives	prototypical AND causatives

Table 6: Types of causative constructions (according to Song 1996, Fig. 5.2, p. 138; reproduced with modifications)

The typology suggested by Song (1996) is quite peculiar since the three types appear to be, first, not uniformly defined, and second, over-generalizing. These drawbacks make it difficult to use this tripartite typology as such. However, the overall attention to implicativeness and grammaticalization was helpful for the current project.

Apart from typology, the volume by Song (1996) offers a critique of the paradigm case (Comrie 1976) and an alternative approach (that also received special attention in an earlier paper Song (1991)). The essence of the alternative approach by Song (1991) is the idea of the "NP density control".

> There seems to be some kind of constraint on the number of core NP arguments allowed per causative sentence in each of the languages. <...> It seems that what these languages are doing is keeping the number of core NP arguments in the causative sentence from exceeding the maximum number of core NP arguments permitted in the ordinary noncausative sentence.
>
> Song (1996, 174)

As one can see, *density control* can be regarded as a different way of saying *demotion*: both processes are concerned with restricting the presence of certain types of constituents within a single clause. However, the above-cited definition by Song (1996) raises two points I would like to highlight regarding the goals of the current study.

First, Song uses the term "core argument" in a sense that is closer to the RRG understanding (contrary to, for example, Dixon and Aikhenvald (2000b), see Section 5.1). According to Song (1996, fn. 13, p. 182), core arguments are "strictly required by the valence of the verb in question". The relation made by this definition to "the valence of the verb" means that the number of arguments depends on the lexical semantics and the diathesis. In RRG, the number of core arguments is determined on the same premises. However, RRG offers more flexibility in handling various morphological encodings. In RRG, there is no one-to-one connection between morphological cases and types of arguments. Hence, the same morphological case can be used for direct arguments, oblique arguments, and adjuncts. In contrast, Song (1996) considers all "oblique" cases as introducing adjuncts only. Thus, while discussing cases of "extended demotion" (in Comrie's terms) and a well-known, very special construction in Nivkh (Gilyak), Song (1996, 176) claims that they contain adjuncts, not arguments. Throughout the present thesis, I nevertheless try to analyze such elements as core arguments in RRG's terms, unless the language data indicates otherwise.

The second important part of the definition of the NP density is its relatedness to non-causative constructions. Song (1996) explicitly refuses to treat causatives as something very special; in contrast, the author claims that causatives must reflect patterns attested in other (non-causative) constructions of the same language. When discussing object doubling in causatives, Song (1996, 177) notices that in languages with this pattern in causative constructions, it is also found in non-causative constructions. Hence, Song (1996, 177) regards object doubling as another way of density control.

The idea of relating causatives not only to the constructions they are derived from but also to other constructions of a given language with the same number of arguments (or participants) is not particularly popular in the literature, but it is crucial for the current study. In fact, identifying the types of constructions to which causatives can be related, from the perspective of using the same marking for the participants, is one of the main goals of this thesis.

3.4.2 Li (2008) critisizing Comrie (1976) and Song (1996)

A relatively short proceedings paper entitled "On the argument realization of causatives" (Li 2008), which is surprisingly close to the one by Alsina (1992), is a curious, relatively recent attempt at defining Comrie's paradigm case, mainly by criticizing Song (1991) and Song (1996). The main accusation by the author is that Song's studies "lack predictive power", and the paper aims to show that "although Song disapproves of Comrie's Case Hierarchy, the Hierarchy is actually full of insight and is indeed needed to account for the case-marking possibilities of causative constructions" (Li 2008, 138).

The author tries to stick to monoclausal causative constructions. First, the author draws attention to the fact that although object doubling is allowed, subject doubling is never attested:

> This demonstrates that when the causee and the base direct object compete for the accusative case of an accusative language and when the former loses,

> the causee cannot bear a case higher than the one for which it is competing. However, it may bear a case lower than the accusative case.
>
> Li (2008, 139)

Building on that, the author suggests an extrapolated "Principle of Demotion", which does not require the demotion to take place exactly like predicted in Comrie (1976) and allows some doubling:

> In forming monoclausal causatives, if A and B compete for a case and if A wins, then B can only bear a case lower than the one A gets; if A and B compete for a case and if the competition ends in a tie, both A and B get the same case. Li (2008, 139)

The author promises to predict all possible argument realizations using this principle and, in fact, provides three variants of causative constructions based on monotransitive verbs. First, the author postulates that the causer and the causee compete for the role of the subject of the derived construction and that the causer always wins. Second, the causee competes with the object of the base verb, which yields all possible – three – outcomes: (1) the causee "loses the competition" and gets demoted to any position lower than DO (IO or oblique), (2) the tie leads to a double object construction, (3) the causee "wins the competition and takes the direct object position" and thus the object of the base verb needs to be demoted somewhere lower.

Unfortunately, this is as far as the paper gets. On the one hand, it offers nothing new, since the limitations of Comrie's analysis and its applicability across many languages are well known. Nevertheless, the observation that a participant cannot occupy a position higher than it competes for is interesting (however vague the concept of *competing for a given position* might seem). In RRG, this concept is somehow built into the Actor-Undergoer hierarchy. Indeed, some participants are more likely to receive a certain macrorole and a specific morphological/syntactic encoding. Therefore, if two participants can be eligible for a given macrorole or syntactic position, they compete. Usually, one of them is located higher in the Actor-Undergoer hierarchy. There are two default case assignment rules that predict INS or DAT on non-macrorole core arguments, but language-specific rules can overwrite the general ones. Hence, RRG is compatible with the approach outlined in Li (2008), with a more elaborate mechanism for handling argument competition at a given position.

3.5 Inspired by Dowty (1991)

3.5.1 Ackerman and Moore (1999): ingredients of the analysis

Ackerman et al. have published a whole series of works dedicated to arguments of causative constructions (Ackerman and Moore 1994; Ackerman and Moore 1999; Ackerman, Malouf, and Moore 2017; to cite just a few). In this section, I present the paper "Syntagmatic and paradigmatic dimensions of causee encoding" that ambitiously suggests a list of concepts that "in some form, need to be present in any adequate account of causee encoding" (Ackerman and Moore 1999, 40). This work crowns prior research by the same team of authors on specific aspects of causative constructions. In general, their works dating back to the 1990s are based on Dowty's approach to arguments in terms of proto-roles (Dowty 1991). This relates their approach to RRG, which also departs from Dowty's findings.

In the conclusion of Ackerman and Moore (1999), the authors suggest the following "ingredients" (authors' term) that they use in their analysis (and suggest using henceforth). They are presented in (12). Below, I present each ingredient in more detail and discuss its (possible) contribution to the present study.

(12) a. A repertoire of semantic proto-properties, rather than atomic θ-roles (Dowty 1991)

b. A Syntagmatic Argument Selection Principle defined in terms of proto-properties (Ibid.)

c. Semantic properties of the causative base predicates

d. A Paradigmatic Argument Selection Principle defined in terms of proto-properties
Ackerman and Moore (1999, p. 40, (52))

Dowty's proto-properties are characteristics of participants that describe their semantics (e.g., *volitional involvement in event or state*) and, to an extent, syntactic behavior (e.g., *incremental theme*). These properties are defined for proto-agents and proto-patients only, which rhymes with the RRG's postulation of two macroroles. Definitions formulated in terms of such properties are more precise than the commonly used definitions of "subject" and "object" (which are totally avoided in RRG) and even of θ-roles (which in RRG tend to be avoided and replaced by more general notions of macroroles).

What is called "Syntagmatic Argument Selection Principle" in (12), is actually Dowty's Argument Selection Principle that relates syntactic encoding to proto-argument properties (Dowty 1991, 576). Ackerman and Moore (1999) call it *syntagmatic* because this principle predicts the argument selection within a single sentence. The general idea of this principle is that the argument with the greatest number of proto-agent properties becomes subject; the argument with the greatest number of proto-patient properties becomes object; with a three-place predicate, the argument having fewer proto-patient properties becomes oblique or prepositional object (Dowty 1991, 576). Later studies have shown that this idea needs refinement, as it predicts only nominative-accusative alignment. Also, languages are known to differ with respect to selecting the second-lowest vs. second-highest participant as an oblique object (see Guerrero and Van Valin (2004) and Guerrero (2007) for an RRG analysis of this specific issue). However, the general idea that all participants that are semantically available for a given predicate receive their syntactic encoding according to their proto-properties seems to hold, with a healthy room for precision.

The novelty of the analysis presented in Ackerman and Moore (1999) lies in the *Paradigmatic* Argument Selection Principle, which cuts across the Syntagmatic principle. Its postulation was motivated by the fact that not all two-argument and three-argument constructions are equal. If the syntagmatic principle alone determined argument encoding, this variation would not occur. Nevertheless, it takes place and, according to Ackerman and Moore (1999), is due to differences in the arguments, namely, their proto-properties. The paradigmatic dimension is thus offered by comparing the arguments competing for the same syntactic slot in terms of proto-properties. The syntactic coding is thus chosen based on these properties.

The paradigmatic principle is exemplified in Tab. 7, where three types of three-place predicates are compared. The question is, how to encode the argument symbolized as 'ARG$_2$'. This slot can be occupied by entities having more or less proto-patient properties. Thus, the most proto-patient argument appearing in this slot will be encoded as a DO, while the least proto-patient argument will be encoded as an OBL.

PRED$_A$	ARG$_1$	**ARG$_2$** **most proto-patient** $\rightarrow$ **DO**	ARG$_3$
PRED$_B$	ARG$_1$	**ARG$_2$** **less proto-patient** $\rightarrow$ **IO**	ARG$_3$
PRED$_C$	ARG$_1$	**ARG$_2$** **least proto-patient** $\rightarrow$ **OBL**	ARG$_3$

Table 7: The Paradigmatic Argument Selection Principle (reproduced from Ackerman and Moore 1999, p. 15, (24))

Like the syntagmatic principle, the paradigmatic principle requires considerable additional precision. However, the general idea that it cuts across the syntagmatic principle explains the systematic inconsistencies in the language structures.

Finally, the remaining ingredient, "semantic properties of the causative base predicates," refers to the incompatibility of certain verbal classes with certain syntactic positions. In other words, these properties shape the range of choices for the argument selection principles.

In my analyses throughout the present paper, I try to pay attention to the principles formulated in Ackerman and Moore (1999). My method of comparing causative constructions to other structures of the same language opens the paradigmatic dimension. Thus, I determine a participant's properties based on language evidence. I believe this method is more reliable than formulating those properties based solely on constraints believed to be universal. However, it is not always easy to trace how exactly the argument selection principles and the constraints on individual predicates and arguments interact in each language. Therefore, I try to offer the most complete account for each case, which is often far from exhaustive.

3.5.2 Li (2020): weighted proto-properties

The paper "Causer and causee as two higher-ranked thematic roles" (Li 2020) is one of the recent additions to Dowty's proto-roles (Dowty 1991). The author understands causation in a broad sense and, in Li (2020), works mainly with Chinese resultatives (which are indeed challenging) and some transitive causative constructions like *The crime scene scared the jogging man* (Li 2020, 612). The author tries to expand their analysis to "monotransitive-based morphological causatives" (i.e., morphological causatives derived from transitive base verbs).

The paper opens by noticing that Dowty's proto-properties are initially formulated as unordered lists. However, weights can be assigned to them so that some properties become more important than others. The author suggests prioritizing the proto-Agent property "causing an event or change of state in another participant" and the proto-Patient property "causally affected by another participant" over others (Li 2020, 602 citing Dowty 1991, 572).

The main claim of the paper is that the thematic hierarchy is updated by postulating two thematic roles – Causer and Causee – at its left (high) end. Thus, the author suggests that in any sentence, the fact of being a Causer is the most important. If a participant does not qualify as a Causer, it might be the Causee. If both these definitions fail, the next eligible thematic role is Agent, then Patient, then other roles. Concerning "monotransitive-based morphological causatives, " this ranking is claimed to hold. However, the mapping of the roles onto grammatical relations is not clear "because of three arguments competing for two core grammatical relations and because of the complexity of determining which argument, in this case, is realized as the direct object" (Li 2020, 627).

The reasoning in Li (2020) is organized similarly to that in Ackerman and Moore (1999), as both papers depart from Dowty's properties, apply them to certain constructions, report difficulties they encounter, and suggest updating the analysis accordingly. What is crucially different between the two papers is that Ackerman and Moore (1999) pays primary attention to proto-Patient properties, while Li (2020) works mainly with proto-Agent properties. For this reason, the papers reach different conclusions regarding the further application of Dowty's proto-properties. In this respect, RRG seems to offer a ground for reconciling the two lines of analysis. In RRG, an Actor-Undergoer Hierarchy is suggested (Van Valin 2005, Fig. 4.4., p. 126), which places prototypical Actors (in Dowty's terms, proto-Agents) at the left extremity and the prototypical Undergoers (in Dowty's terms, proto-Patients) at the right extremity. Participants with mixed proto-properties are listed between them. RRG acknowledges that languages, or even individual rules within a language, can count participants in either direction, from Actors to Undergoers or from Undergoers to Actors. This approach pays equal attention to all the properties and controls their weights and/or relevance for applying a given rule.

Although the general idea of treating the proto-properties not equally but weighted looks pretty productive, the conclusions by Li (2020) do not help much to study three-argument causative constructions. RRG offers a more powerful and versatile system of using Dowty's findings in this respect.

3.6 Studies of the causee

3.6.1 Alsina (1992): attention towards the argument structure

The study "On the Argument Structure of Causatives" (Alsina 1992) is realized within the Lexical-Functional Grammar (LFG) framework. It aims to suggest an analysis opposed to the previously expressed theories "that derive causatives through syntactic incorporation, particularly those of Baker (1988) and Li (1990)" (Alsina 1992, 517). The primary goal of this novel theory is to explain object doubling in Chichewa (Bantu; N31 in Maho (2009); see my analysis of Bantu languages in Section 6.2.1). Its second goal is to add a semantic dimension (which seems to be a synonym of "argument structure" in the paper) to the study of causatives, which had previously been purely syntactic.

The study starts with presenting the data from Chichewa that demonstrates an alternation between two causative strategies: the one where the causee is introduced by a preposition "elsewhere used to introduce goals and directional locatives" and the other one, where it is an object (which leads to doubling). Having explored the data, Alsina (1992, 520) formulates the following goals for studies of causatives: "A theory of causatives must account for [1] what it is that allows the argument that corresponds to the external argument of the embedded verb to be expressed as an object in the causative form, [2] what it is that allows it to be optionally expressed as an oblique or omitted, and [3] what it is that allows this optional oblique expression only for some verbs" (numbers are mine – VG). Working in the paradigm of LFG (Lexical-Functional Grammar), the author suggests a theory "which relates semantic arguments to their morphosyntactic expression by means of syntactic function" (Alsina 1992, 544). This is particularly interesting for the goals of the present study, as I seek to propose a similar theory within RRG.

First, the author suggests analyzing the causative morpheme as a three-place predicate, its participants being (1) the causer, (2) the individual affected by the causer (the causee), (3) the event caused by the causer and of which causee is a participant (Alsina 1992, 521). Curiously, this representation describes both causatives of transitives and causatives of intransitives, postulating the same three-place nature of the causative morpheme. This analysis is supported by the mention that "verbs meaning 'give' are used as causatives in various languages" (Alsina 1992, 522). This being only one of the possibilities for building causative constructions, the idea of analyzing all causatives as three-place predicates is very appealing.

The author highlights that there is one participant that belongs to both the CAUSE predicate and the embedded verb (the one from which the causative is derived). According to the author, the linguistic variation in argument encoding stems from the variety of possibilities for arguments to *fuse*, i.e., for different arguments of the embedded verb to be shared with the arguments of the CAUSE clause. Thus, the author describes the mechanism leading to encoding causees as adjuncts.

> The option of realizing the causee of certain transitive base verbs as an adjunct comes about when the patient of the causative predicate is not fused with the causee. In this situation, the causee, as a noninternal argument, is not assigned the syntactic expression that corresponds to an internal argument. Because it is not the external argument of the derived argument structure either, it cannot be the subject; hence, it receives no syntactic expression as an argument and can only be realized indirectly as an adjunct.
> Alsina (1992, 522)

After having explored Chichewa causatives, the author suggests applying to this data the "lexical mapping theory" initially developed for other phenomena in Bantu languages (Bresnan and Moshi 1990; Alsina and Mchombo 1990 *inter alia*). This theory suggests assigning the features 'restricted' and 'objective' to arguments and applying three mapping principles to determine their syntactic realization. In other words, these features and principles combine semantic roles with the syntactic coding layer. This theory explains, in a very illustrative way, the alternation of two causative constructions within the same language: one encoding the causee as an object and the other using a PP for that purpose. Nevertheless, it does not specify *what kind* of adjunct should be used, nor does it account for other positions for the causee realization. Thus, although interesting and even rhyming with RRG in its basic idea of using semantic information to analyze syntactic encoding, this theory is barely scalable to languages other than Bantu.

3.6.2 DeLancey (1984): an early prototype approach to features

The paper "Notes on agentivity and causation" is also more of a museum-like knowledge today; in addition, it is not very close to the topic of my work. However, it is an illustrative example of a prototype approach to linguistic features, which can be useful to date. In short, the prototypical approach defines categories by establishing a list of features that demonstrate elements of this category. Items that totally conform to the definition are called *prototypes*. However, items that lack a complete list of features are not excluded from the category; instead, they are pushed to its periphery, i.e., further from the prototype. More details on the initial prototype theory can be found in Rosch (1978) and the consequent debate. See also works by George Lakoff, where the theory of prototypes is applied to study linguistic concepts and combines with methods of cognitive linguistics (Lakoff 1977; Lakoff 1982; Lakoff 1984 – just to name a few; Lakoff 2007 for a fairly recent update).

The paper examines two-participant sentences that involve at least some semantic component related to causation. In some of these sentences, the causers look like agents (these are prototypical transitive clauses). However, in some others, causers are semantically and syntactically different (e.g., *He was killed by the lightning*., DeLancey (1984, 208)). The investigation shows that both the "properties inherent in the entity" as well as the "aspects of the overall causation schema" determine whether a given participant in a given language is treated as an agent or not (DeLancey 1984, 206). Thus, the somewhat novel for the time idea of "not a perfect exemplar" (DeLancey 1984, 208) of an agent is expressed.

One of the major goals of the paper is the formulation of a prototypical transitive event:

> The prototypical transitive event is one that can be traced back to a single cause from which an unbroken chain of control leads to the effect.
>
> DeLancey (1984, 207)

Technically, this definition does not specify the number of participants in the mentioned chain and thus presents transitivity in a more mathematical sense. The control feature is nevertheless relevant and will be addressed in many further studies on agentivity and causality in a broad sense.

The relevance of DeLancey (1984) seems relatively minor for my current work. It does not tackle three-argument sentences, and the account of various types of causes offered in Nedyalkov and Sil'nitskiy (1969a) is more powerful and covers the cases shown in DeLancey (1984) as well as many others. However, the prototypical turn of the debate is worth remembering in theoretical and typological studies. In my work, I determine features relevant for building different argument structures in my data, including those of "entities" and "the overall causation schema". However, I do not rank them in terms of being necessary for a prototype. Nevertheless, such a ranking can help explain some of the variation. On the other hand, nowadays, a prototype-inspired study should perhaps rely on quantitative data or at least take such data into account. Since I am not working with a typologically representative sample, I might not have enough data to formulate a definition of a prototype. Hopefully, these studies are yet to come.

3.6.3 Anderson (2005): features instead of grammatical relations

The paper "On the argument structure of morphological causatives" (Anderson 2005) is an attempt to explain the argument structure of causative constructions without resorting to the notions of *direct/indirect object* and other grammatical relations. Instead, it suggests representing semantic roles as combinations of features, which, in turn, would determine the morphological coding of the NP in a sentence (the paper deals exclusively with dependent-marking languages). The feature telling that the participant is agentive is called `erg`; the feature roughly corresponding to Fillmore's *objective* (Fillmore 1967) is called `abs` (Anderson 2005, 38). Another feature widely used throughout Anderson (2005) is `loc`. The author first formulates two patterns claimed to cross-linguistically occur in constructions with verbs of transfer (see original representations with corresponding English examples in Tab. 8). Accordingly, the author formulates two causativization rules yielding these two patterns. In any case, the causativization adds a new `erg` argument, which corresponds to the causer in common terms. The former `erg` argument, i.e., the agentive participant of the base verb, either receives an additional `loc` feature (to become `[erg, loc]`) or gets its `erg` feature replaced by the `loc` feature. The first type of causative derivation thus results in having an `erg`, an `[erg, loc]`, and an `abs` – like the *receiver* strategy in Tab. 8. The second type of causative derivation results in having an `erg`, a `[loc]`, and an `abs` – like the *oblique* strategy in Tab. 8.

Pattern name	Argument structure	English example
receiver	`[erg] [erg, loc] [abs]`	Fred gave his aunt a present.
oblique	`[erg] [loc] [abs]`	Fred gave a present to his aunt.

Table 8: Argument patterns attested with verbs of transfer (reproduced from Anderson 2005, p. 50, (45a, 45b))

The approach suggested in Anderson (2005) is close to the initial idea of RRG in abandoning traditional notions of grammatical relations. The definitions of the `erg` and `abs` features correspond to the RRG's concept of macroroles. However, RRG has a more elaborate system for dealing with non-macrorole participants and, generally, for handling matters more complex than simple transitive patterns. This makes Anderson's approach less useful in the presence of RRG.

Another point of the paper Anderson (2005) is to relate morphologically derived causative constructions to (presumably underived) patterns for transfer verbs. I find this general line of analysis insightful and build my work following it. However, Anderson (2005) claims that there are exactly two cross-linguistic patterns. In my work, I am not ready to make such generalizations right away. Thus, I first compare different argument patterns within each language to postulate a more general abstraction afterward.

3.6.4 Saksena (1980): affectedness matters

The paper "The Affected Agent" (Saksena 1980), once translated into more modern, formalized terms, aims to demonstrate the relevance of the feature 'affected' for participants that can be semantically defined as agents. It is also one of the early attempts to explain the co-existence of several argument-marking strategies in a single language (namely, Hindi). The paper opens with a presentation of Hindi causatives able to mark causees either with *-koo* (traditionally glossed as DAT/ACC) or with *-see* (traditionally glossed as INS). It is nice to see the comparison of these markers across several Hindi constructions throughout the paper. The main part of the paper lists semantic and syntactic effects and tests whether they hold across the causative constructions. These effects show a systematic difference between participants receiving *-koo* vs. *-see* marking. Thus, the author claims that *-koo* marks causees that are affected, while *-see* marks causees that are non-affected, and that the affectedness determines their syntactic behavior. Afterward, the author provides a second list of syntactic situations in which the NPs marked with *-koo* and *-see* exhibit different properties, which the author explains in terms of their difference in affectedness.

The paper "The Affected Agent" is useful for my present work in many respects. First, it is a valuable source of Hindi examples and some interesting remarks about them. This has especially influenced my reasoning in Section 6.1.1.5. Second, it provides a systematic comparison of two causative constructions with different causee marking, thereby providing a direction for reasoning about their coexistence. In classical RRG, the competition between these strategies is explained from a discourse/pragmatic perspective, but the underlying difference remains related to affectedness. Third, the paper Saksena (1980) extensively compares causative and non-causative constructions in Hindi. This reasoning is easy to reuse in other work, as it operates with concepts and reaches conclusions that map onto other descriptive findings and generalizations.

3.6.5 Wali (1981): causees as second-order agents

The paper by Wali (1981) is remarkable in several respects. First, it begins with a confident and extensive critique of the approach presented in Comrie (1976), which was considered very novel and powerful at the time. Second, it appeals not only to modern languages but also to historical data from Latin and Sanskrit. At the end of the paper (Wali 1981, 306–307), Sanskrit is used as a testing ground for hypotheses about the diachronic development of the causee. Third, and most importantly, the paper focuses on the problem I outline myself in Section 5.1: the marking of the causee.

After presenting the failure of constraints elaborated by Comrie (1976) for some languages, Wali (1981, 294) shows a range of examples from different languages, where causees of three-argument constructions are obligatorily not marked like indirect object (= recipients in ditransitive clauses). (This phenomenon is central to Chapter 6, where I try to list these varieties.) Wali (1981, 294–297) also discards the idea that causees sometimes repeat the marking of agents in passive clauses due to some kind of transformation. A rich theoretical argumentation and analysis of examples from varied languages support this refusal. In contrast, Wali (1981, 294) makes a claim about the causee that "in all the cases without exception is that it invariably centers around the concept of a second order agent, its form ranging from a demoted passive agent to an instrumental or some other oblique agent NP. In other words, a deep resemblance to demoted (second-order) agents is emphasized. Unfortunately, the paper's format has prevented the author from presenting his proofs of this generalization. Nevertheless, the data I examine also contribute to the proof of this hypothesis, making a well-formed proof the next step in this research direction.

After capturing the similarities between causees and agents, Wali (1981, 297) questions whether they resemble objects, especially indirect objects. The answer is that "languages consistently differentiate between the causee IO and the true IO" (Wali 1981, 297) and "that the behavioral features of the causee do not always correlate with those of the true IO" (Wali 1981, 300).

Merging the two questions, Wali (1981, 300) comes to the following conclusion: "The cross-linguistic data suggests that the causee is some type of second order agent and is only masquerading as an IO in some languages, in so far as it shows the low level surface coding property of an IO". One could interpret this as an attestation of a discrepancy between the causee's morphological and syntactic (plus semantic) properties. A further hypothesis in this direction would be that using IO-like marking for causees is a minimally complicated strategy since ditransitive constructions already have a device for encoding a third argument. I elaborate on this in Chapter 6.

The second part of the paper by Wali (1981) is dedicated to the semantic analysis of the causee's case. It also starts with a disproof of a better-known approach, namely the one suggested in Cole (1976). This approach predicts that the causee's role depends on its agentivity. However, Wali (1981, 306) concludes that "it is the abstract contactive/non-contactive causation that determines the status of the causee and not the agency". It is explained that in a contactive causative situation, the causer directly performs the action, making the causee "a mere patient" on the semantic level and morphologically marked as an object. In the opposite situation, when the causer is "a super-planner or a director", the causee's role and marking "is modified accordingly".

This part of the analysis has much in common with the discussion of the RRG concept known as *agentivity implicature* (Holisky 1987; Van Valin and Wilkins 1996; Van Valin 2018a), which cares about whether the causer (sic!) can be construed as an effector or not. In the analysis by Wali (1981), the role of the causer and the type of causation are not separated. A deeper look into their relationship is undertaken in Chapter 6.

To sum up, the paper by Wali (1981), although quite concise and rarely cited in the first row of causative-related works, is crucial to the present study and significantly influences the reasoning.

3.6.6 Cole (1983): causees retain their agentivity

Peter Cole has dedicated much of his work to causative constructions. For this review, his paper "The Grammatical Role of the Causee in Universal Grammar" was chosen because it is his most concise work on the question central to my work, summarising prior findings.

The paper examines "clause union causatives," but the examples show that morphological causatives, such as those in Hebrew and Kannada, also fall under this definition. The central claim of the paper is that "the grammatical role in derived structure of the underlying complement subject is determined by the semantic role of that NP in the complement clause" (Cole 1983, 116). In simpler terms and with tangible examples, the following implication is drawn. Suppose the subject of the base (transitive) verb is agentive and retains control over the situation, in a three-argument causative construction. In that case, it will receive "a grammatical role associated in the language in question with agency or animacy". Cole (1983, 117) specifies that these roles are most frequently "instrumental case and indirect objecthood." A touching and quite relevant note for the present study is found on Cole (1983, 125): "instrumental case (the same case used for passive agents)". Even though it is not completely accurate from a cross-linguistic perspective, it captures the relation that I find very important.

In the section "Principles determining the selection of grammatical role" (Cole 1983, 125), which is way too short in contrast to the expectations given by the title of the article, the author emphasizes that the agentivity of the causee is the one and only relevant semantic criterion for determining its grammatical role. Consequently, multiple ways of causee marking in a single language must reflect the agent vs. non-agent contrast. By giving two examples of contrast, the author seems to overlook one that is frequently cited and attested across the world's languages: dative vs. instrumental.

As shown by Wali (1981), this approach (also couched in earlier works by Cole) does not hold for many languages. The data I have (see Chapter 6) also witnesses the inaccuracy of Cole's claims. The lack of attention to the differences between instrumental and dative causees makes the approach even less applicable to data from other languages.

3.6.7 Horvath and Siloni (2011): a particular approach to demotion

In the paper "Causatives across components" (Horvath and Siloni 2011), the authors draw the readers' attention to the fact that not all morphological causatives have the same syntactic properties. Namely, in some languages, morphological causatives resemble syntactic (analytical, periphrastic) ones, while in other languages, morphological causatives are more like lexical ones. The difference is thus in the status of the causative morpheme: either it acts more like a separate predicate or modifies the base predicate. To distinguish between these two types of morphological causatives, the authors suggest several diagnostics (Horvath and Siloni 2011, 665–672), including the scope of negation, the ambiguity of the VP-ellipsis, anaphor binding, the applicability of agent-oriented adverbials to the causee, raising, and control.

This point is crucial to my work. The argument structure of sentences with multiple predicates is very different from that with a single predicate, since the constraints on the uniqueness of role realization and marking no longer apply. In my work, however, I am interested in the situations where these constraints nevertheless apply. Therefore, I had to exclude some languages from my sample since there is evidence that they have morphological causatives with multiple predicates. In particular, this is the case of Japanese, which is cited as an example of a language having syntactic-like morphological causatives in (Horvath and Siloni 2011).

When discussing morphological causatives that are "lexical", i.e., contain a single predicate with an additional participant contributed by the causal morpheme, Horvath and Siloni (2011) conceptualize their arguments in terms of two features, following Reinhart (2002). The first feature is `[± c]`, it "determines whether or not the argument in question is necessarily responsible for causing the denoted event" (Horvath and Siloni 2011, 677). This feature is claimed to be positive for the causer, and there appears to be a restriction on multiple positive values of this feature within a sentence. Thus, according to the final definition of a lexical causative offered in Horvath and Siloni (2011) (see (13)), if any argument of a base verb has a `[+c]` feature, its value changes to `[-c]`.

(13) $V < \alpha > \rightarrow CAUS - V < [+c + m] \ \alpha' >$, where α includes a role specified as external; if this role includes a `[+c]` feature, the feature is revaluated to `[−c]` (otherwise α equals α'). Horvath and Siloni (2011, p. 692, (62))

The second feature is `[± m]`, "which determines whether or not the mental state of the argument in question is relevant to the denoted event" (Horvath and Siloni 2011, 677). The distribution of the values of this feature is somewhat tricky. As for the causer, the authors say that it "must be animate (its mental state is relevant)" (Horvath and Siloni 2011, 677).

The authors make an important note concerning the `[−m]` feature assigned to participants having the `[+c]`, see below.

> At the semantic interface, an argument bearing a `[+c]` cluster, which is unspecified with regard to `m`, can be interpreted either as an Agent (*Dan* in (14)) or as a non-Agent (say, an Instrument, *this key* (14)) (see Marelj (2004)). The mental state of an argument specified `[−m]`, say `[−c−m]`, the door in (14), remains irrelevant at all stages of the derivation, including the semantics.
>
> (14) Dan / This key opened the door.
>
> Horvath and Siloni 2011, 678

In RRG, the feature `[c]` by Horvath and Siloni (2011) corresponds to the notion of the effector. Indeed, causers are effectors, and effectors, overall, are in control of bringing about the event. The alternation between `[+c +m]` and `[+c −m]` roles is captured by the RRG rule for assigning instrumental case (Van Valin 2005, p. 110, (49)): if a clause has two arguments concurring for the same macrorole (i. e., are `[+c]` in accusative languages/constructions) and one of them is higher on the Actor-Undergoer Hierarchy (i. e., one is `[+m]` and the other is `[−m]`), then the highest receives the (Actor) macrorole, and the lower receives instrumental case. The classic example of such a sentence would be *Dan opened the door with this key*. The promotion of the instrumental argument to the PSA (privileged syntactic argument $\approx$ syntactic subject) is an alternation available for effector, i.e., `[+c]` arguments.

Overall, it seems that the values of the `[m]` feature are determined by the base verb and are not affected by the causation. This is how the authors comment on the assignment of feature values to the participants of the Hungarian verb *olvas* 'CAUS-read': "Its input's Agent, namely the causee, performs the reading event and therefore its mental state is relevant, but it is not the one that triggers or brings about the event of reading. In feature terms, this means that the valuation of its `c` feature becomes negative". Moreover, when performing a contradiction test (16) on Hungarian sentences like (15), the authors deduce the entailments summarised in (17). These entailments can be summarised even further: the causativization does not affect the values of the feature `m`.

(15) János meg-et-et-te Mari-val az almá-t HUNGARIAN
János.NOM PRF-eat-CAUS-PST.DEF.OBJ Mari-INS the apple-ACC△
'János made Mari eat the apple.' Horvath and Siloni 2011, p. 679, (42a)

(16) a. *de nem Mari et-te meg az almá-t HUNGARIAN
but NEG Mari.NOM eat-PST.DEF.OBJ PRF the apple-ACC△
'but it wasn't Mari who ate the apple'

b. *de Mari nem az almá-t et-te meg
but Mari.NOM NEG the apple-ACC eat-PST.DEF.OBJ PRF△
'but it wasn't the apple that Mari ate Horvath and Siloni 2011, p. 679, (43)

(17) a. `[−c −m]` of the CAUS-V has the entailments of `[−c −m]` of the corresponding V.

b. `[−c +m]` of the CAUS-V has the entailments of `[−c +m]` of the corresponding V.

Thus, the authors suggest the following semantic representation of (15) (and of other similar constructions), following the notation known from Parsons (1990) (commonly referred to as the *neo-Davidsonian*):

(18) ∃e [CAUS-eat(e) & [+c +m](e, János) & [-c +m](e, Mari) & [-c -m](e, the apple)]
Horvath and Siloni 2011, p. 679, (42b)

Yet the feature names used in Horvath and Siloni (2011) seem quite novel, the idea they reflect is well known. The feature `[m]` tells that the actor and the undergoer of the base predicate keep their relation to each other, or, in other words, that the (macro)role structure of the base verb is intact by the causativization. The change of the value of the feature `[c]` assigned to the causee reflects some kind of demotion – again, a well-known phenomenon. The causee is also the only participant that has both positive and negative feature values. In fact, this duality captures well the fact often called "being the shared argument". Indeed, the causee is related to both the causer (along the `c` axis) and the affectee (along the `m` axis).

This account, however, does not explain the variety of causee encodings. Perhaps, it could predict any difference related to verbal classes. If one hypothesized a language with a transitive verb, whose both participants are `[−m]` (i.e., the mental state of both participants is not relevant for the event), then one would need to postulate the causative representation like (19).

(19) ∃e [CAUS-V(e) & [+c +m](e, causer) & **[-c -m](e, causee)** & [-c -m](e, affectee)]

Hence, some verbs yield causees with the `[-c +m]` feature structure, and others yield causees with the `[-c -m]` feature structure. Nevertheless, there are more than two options for causee coding across languages, so this analysis is not powerful enough to capture all of them.

3.7 Kemmer and Verhagen (1994): situating causatives in a language

The paper "The Grammar of Causatives and the Conceptual Structure of Events" is crucial to the present work because it influenced and shaped the methodology I used to describe the data. Belonging to the field of cognitive linguistics, this paper offers "an explanatory semantic account of the variation of case markings found in causative constructions" (Kemmer and Verhagen 1994, 115). Not only does it work with the same entities of linguistic reality as I do (case marking is the major although not exclusive strategy of argument marking), but it also suggests the same perspective of explaining the attested variation: "it [the approach] predicts that such variation is related systematically to the semantics of case markings in simple clauses, which is in fact the case" (Kemmer and Verhagen 1994, 115). Nevertheless, case markings are not detached from the interpretations of causative constructions as a whole: "The marking of the causee is a consequence of conceived differences in its role in the causal event, which relate to such aspects of event structure as (in)directness of causation and (relatedly) degree of agency and affectedness of participants" (Kemmer and Verhagen 1994, 115). I could never advocate for this semantically-based approach to the study of causative constructions better than the authors do themselves:

> This approach not only has wider empirical coverage than syntactic, hierarchy-based accounts but is simple, unified, has greater explanatory power both for cross-linguistic variation and for intricate intralinguistic distributional facts; finally, it accords with a cognitively-based view of language, in which the knowledge underlying grammar is not qualitatively different from other aspects of human understanding and reasoning.
>
> Kemmer and Verhagen (1994, 115)

The paper opens by opposing itself to "a common approach" in treating causative constructions not as "derived by reduction" from biclausal structures, but "as built up from simpler structural/conceptual units" (Kemmer and Verhagen 1994, 116). Another important delimitation is that the authors "are not referring to some notion of causation in the physical world, but rather to the human conceptualization of causation" (Kemmer and Verhagen 1994, 117). This general idea excludes simple transitive verbs, which seems natural within a semantically motivated approach. The authors restrict their definition even further by postulating the requirement of the absence of any specific lexical meaning on the causal predicate: "All that is expressed by the predicate representing the causing event, which I will call the "causal predicate", is the pure notion of cause (or the closely related one of enablement; cf. below), without more specific lexical content" (Kemmer and Verhagen 1994, 117). As the paper shows, not only enablement but also permission are regarded as possible extensions of the pure causal predicate. In this respect, the delimitation made in Kemmer and Verhagen (1994) is slightly looser than in Nedyalkov and Sil'nitskiy (1969a).

Speaking of the arguments of a causative construction, the authors define them as follows:

> - The causer <...> is the entity viewed as causing the entire event.
> - The causee is the entity carrying out the activity designated by the affected predicate.
> - <...> The affectee, where present, is the entity that is the endpoint of the energy (literal or metaphorical) expended in the entire causative event.
>
> <...> We take these roles as theoretically significant, rather than as convenient labels, and will claim below that they are more crucial than any roles selected by individual predicates in the causative construction.
>
> Kemmer and Verhagen (1994, 119)

The terms *causer* and *causee* seem to be fixed across linguistic approaches, while the term for the third participant, *affectee*, is subject to variation. It is often called *direct object* or simply *object* because it is most often encoded as an object. When it is not the case or when emphasizing the derivation of a causative construction from a base verb, the term *former (direct) object* is often used. Neither are these labels convenient, leading to potential confusion of participants belonging to the different types of constructions, nor are they theoretically salient, as not all affectees are objects or bear an object-like marking. Another tradition is to use the term *theme*, which somehow relates causative constructions to ditransitive ones (featuring *agents*, *recipients*, and *themes*. This labeling tradition is preferable to the one using the term *(direct) object* because it does not imply any specific encoding for the participant in question. However, if causatives and ditransitives appear in the same discussion, the term *theme* still might lead to confusion. Therefore, in my work, I use the term *affectee* in the sense it is used in Kemmer and Verhagen (1994) to have a unique label for this participant of the causative construction and to eliminate any presupposition of its resemblance with any other participant of any other construction.

Speaking of the semantics of causative constructions, Kemmer and Verhagen (1994, 120) suggest three oppositions:

1. physical vs. non-physical,
2. direct vs. mediated (the term *indirect* is used as synonym of *mediated*),
3. cause *per se* vs. enablement and permission.

Theoretically speaking, these oppositions should yield eight types of combinations. The paper, though, lists only three options: *direct physical*, *indirect physical* and *inductive* ("the causee is an agentive entity who is caused to act by some nonphysical means, typically by some verbal Stimulus produced by the causer"), which seems to denote both *direct* and *indirect non-physical* causation. The dimension "cause vs. enablement/permission" does not characterize how causation is realized. Instead, it distinguishes two semantic options for the act of causation.

> A fourth type, *enablement/permission*, involves not the exertion of force on an entity to bring about an event that otherwise would not have happened but the removal by the causer of a conceived barrier that was preventing the causee from carrying out or undergoing the effected event. *Enablement* refers to the case where the barrier is physical <…> and permission to the case where the barrier is social or sociophysical in nature <…>; we can thus consider enablement and permission as subvarieties of one type.
>
> Kemmer and Verhagen (1994, 120)

In fact, these three oppositions correspond well to the two oppositions drawn by Nedyalkov and Sil'nitskiy (1969a): the first two map onto "contact vs. distant" and the third one reflects "factual vs. permissive". Interestingly, Nedyalkov and Sil'nitskiy (1969a) also do not offer subtypes of *permissive*, suggesting that this opposition's perception as a different dimension is deeply rooted in linguistic reality.

In the authors' approach, these semantic differences complement the analysis to capture the variation of case marking. While reviewing Comrie's suggestion of the paradigm case (Comrie 1976), the authors draw attention to the fact that it does not predict any causee marking variation with a given base verb. However, this variation is attested in the language data. Thus, they suggest that "supplemental principles would be required to account for such differentiation" (Kemmer and Verhagen 1994, 124).

Thus, the authors approach argument marking in causative constructions from the perspective of them corresponding to a "structural and conceptual template" (Kemmer and Verhagen 1994, 125) existing in "simple clauses" of a given language. Concerning causatives of intransitives, the authors conclude that they most often correspond to prototypical transitive events from the semantic point of view (this question is not central to their work or mine). As for causatives of transitives, the authors share the opinion "that the most prototypical ditransitive verbs are lexical causatives" and "involve the idea of an agent causing something to happen, for example, causing an entity to come into possession of something, causing an entity to be located somewhere, causing an entity to come into view, etc." (Kemmer and Verhagen 1994, 128). The instrumental construction is regarded as a viable structural correlate as "both instruments and causees in TCs[8] <…> are causal intermediaries on the path of energy flow elaborated by the clause", "an animate entity can be viewed as a kind of metaphorical Instrument, employed by the causer to get an action carried out" (Kemmer and Verhagen 1994, 129).

In the present work, I extrapolate this method to explain other variants of causee (and affectee) realization. I suggest using non-causative constructions of a given language involving participants with the same morpho-syntactic properties to shed light on the properties of the participants in causative constructions. The difficulties arising while applying this method are discussed and compared with some ideas by Kemmer and Verhagen (1994) in Section 5.4.

What is also well observed in Kemmer and Verhagen (1994, 130) is the place of the causee as opposed to both causer and affectee: "[t]he causee in a TC construction, whether dative or instrumental, shares with its simple clause counterparts the property of being in some sense a "third" participant, less crucial to the structure of the event as a whole than the subject and direct object, which are conceptually the core participants in the event <…> it is still a participant that directly interacts with the other two at the level of the causal event as a whole, just as indirect objects and instruments interact directly as participants with respect to a simple predicate". To support and develop this argument, RRG is especially helpful with its distinction of direct vs. oblique core arguments (see Section 2.5.1). Namely, the notion of an oblique core argument captures both the "direct interaction" with the whole CORE (event structure) and some degree of being "less crucial". However, I would like to argue that the question of whether the causee is a direct or an oblique argument should be answered specifically for each language. My analysis in Part II thus develops along this line.

Another idea expressed in Kemmer and Verhagen (1994) that rhymes almost perfectly with RRG devices is the suggestion to represent grammatical constructions "as schemas (of different degrees of specificity), some of which can be used as the basis for extension for the formation of more complex grammatical patterns" (Kemmer and Verhagen 1994, 145). The schemas are suggested to be built from scratch for simple constructions, and those for more complex phenomena would be derived from the simple ones. As shown and acknowledged by the authors, "viewing causative constructions as schemas in the above sense allows a unified account of causatives" in different languages. This is one of the aims of a sound cross-linguistic study and of a theory that

[8] causatives of transitives, foontone mine – VG

aims to describe data from multiple languages.

This notion of a schema resembles very much (including the naming) the RRG concept of "constructional schemas" which are basically schematic representations of constructions (see details in Section 2.7). In RRG, these schemas include information relevant to all levels of linguistic analysis for describing a given construction. Constructional schemas can derive from each other, especially when written in a formalized form, for instance, modeled using the XMG framework (as in the present work, see Part III.

I engage with the data discussed in Kemmer and Verhagen (1994) and discuss their interpretations in Section 5.4, as they have strongly influenced the development of my method for working with causatives-of-transitives across different languages.

3.8 Shibatani and Pardeshi (2002) and De Busser (2010): features and dimensions

The study "Beyond the causative continuum"(De Busser 2010) builds directly on the first one, "The causative continuum" (Shibatani and Pardeshi 2002). Even though neither of these works is as influential as many papers mentioned throughout this chapter, the ideas expressed there are very consonant with my work's most important methodological and theoretical turn.

In "The causative continuum," the authors argue against two popular categorical scales for describing causative constructions. Although it mainly deals with causatives of intransitives (with derived transitives), the general approach can be easily scaled to cover additional phenomena. The first scale is direct vs. indirect causation. After discussing the terms and suggesting the use of the most accurate formulations, the author presents examples of causative constructions that do not fall into either of these two types. These are entitled "sociative causatives" since, in most of them, "the causer performs the same action as the causee" (Shibatani and Pardeshi 2002, 148). As these constructions involve two agents, they can be partly related to the indirect type. At the same time, they also have an overlapping "spatio-temporal" pattern, i. e., the requirement for the causation and the caused action to happen at the same place and time. The author supports the "reality and significance of the proposed intermediate category" by drawing attention to the fact that in "a fair number of languages" causative morphemes also serve as applicative ones and introduce thematic roles usually associated with this type of verbal derivation (comitative, instrument, beneficiary) (Shibatani and Pardeshi 2002, 166). These are also semantic components often appearing in sociative causatives. Thus, the existence of this third type proves that the binary distinction between direct and indirect causatives can be imprecise and that a "tripartite classification" is more relevant (Shibatani and Pardeshi 2002, 153).

The second classification suggested to be reconsidered is the “formal” one, traditionally comprising lexical, morphological, and analytical (periphrastic) constructions. The author points out that these types reflect the “degree of grammaticalization” (Shibatani and Pardeshi 2002, 153), which is a running process where the phases are not necessarily discrete.

The talk by De Busser (2010) goes “beyond” these assumptions and suggests that “causation is part of a much larger n-dimensional functional field” (De Busser 2010, 10). In other words, more parameters condition the variation of causative constructions, each representing a continuum. The author suggests using a modular approach to decompose complex phenomena, namely, decomposing the feature of Agentivity into Agency and Control (De Busser 2010, 8–9).

The idea motivating this way of analysis is expressed at the beginning of the paper and seems very important to my study:

> In this talk, I will focus on causatives and demonstrate how their perceived importance often obscures the fact that in some languages, there is a wider spectrum of morphological or syntactic mechanisms that are structurally and functionally very similar to causatives, but are mostly ignored in descriptive and typological studies. De Busser (2010, 1)

In my work, I also seek to look at the broader picture by decomposing it into smaller modules.

The general idea of a formal continuum is well treated in RRG. Moreover, all nexus juncture types (see Section 2.6) combined, not a tripartite, but a more elaborate scheme is predicted. The idea of assigning features to all necessary components meets the requirements expressed in De Busser (2010) for using smaller information modules to describe causative phenomena. The causative-applicative syncretism mentioned in Shibatani and Pardeshi (2002) can also support the idea of causatives being part of a larger field expressed in De Busser (2010). RRG, as a theoretical framework, offers a top-down (from theory to evidence) approach that can predict which complex units are related. In my work, I try to relate causative constructions with other constructions in a given language to explore as many vectors in this n-dimensional field as possible. As one of the outcomes of the Part II, I list those small modules relevant for tracing these relationships, which I guess is quite in line with methods suggested by De Busser (2010).

3.9 Kozinsky and Polinsky (1993) on coding conflict

The ample volume, edited by Comrie and Polinsky (1993), entitled “Causatives and Transitivity”, is designed as a tribute to the Leningrad Typology School. The content reflects this goal: the volume unites studies inspired by similar ideas and methodologies applied to very different language phenomena.

Many articles investigate constructions with causal semantics (causative constructions in a broad sense as defined in Nedyalkov and Sil'nitskiy (1969a)) in relation not only to other verbal derivations affecting the argument structure (passive, adversative and other voices, as well as transitivity increase) but also to other types of sentences (e. g., Podlesskaya (1993) investigates the role of causative meaning in Russian argumentative, implicational, controlling and estimative sentences).

The volume also contains an influential work by Martin Haspelmath (Haspelmath 1993), which deals with pairs like (20). Since three-argument verbs are not involved in these inchoative/causative alternations, this work has no direct influence on the present thesis.

(20) a. rasplavit' vs. rasplavit'-sja — Russian
'melt (tr.) = cause to melt' vs. 'melt (intr.) = become melted'

b. xajl-uul-ax vs. xajl-ax — Khalkha Mongolian
'melt (tr.)' vs. 'melt (intr.)' — Haspelmath 1993, p. 89, (2)

In the present study, the contribution by Kozinsky and Polinsky (1993) is of particular interest. In this paper, the authors tackle the problem of direct object doubling in three-argument causative constructions. In fact, this is a widespread situation in many genetically unrelated language groups, which is quite tricky to handle with the traditional approach to causees expressed in Comrie (1976).

The authors attest that they are working in the framework of Relational Grammar, yet with some deviations. Not to mention the usage of less theory-specific terms, the authors restrain themselves from using theoretical generalizations not supported by the language data: "We assume that if two categories of nominals are not distinguished in a given language, then they bear the same grammatical relation in this language" (Kozinsky and Polinsky 1993, 181). This turn also rhymes with the approach of the Leningrad Typology School and the present study.

The central part of the paper presents an extensive analysis of causative constructions with doubling in two languages: Korean and Dutch. Even though the Dutch data represents an infrequent situation and the authors present a comprehensive analysis of its peculiarities, it must be discarded in the present study. The reason is that the construction in Dutch is not a morphological, but a compound causative. As for Korean, it is an illustration of a type of language where the initial direct object is moved to a "non-term" (corresponding to "off-core" in RRG terms) position.

In both languages, the doubling appeared to be morphological but not syntactic. The idea of delimiting these two parameters originating from Kozinsky and Polinsky (1993) plays an important role in composing the new typology of three-argument causative constructions in Chapter 6.

One of the most convincing parts of the work is Section 5, where the authors present the tests they conducted to obtain the results described for both languages. This methodology inspired the proofs for claims concerning the number of arguments in the present study.

3.10 Selected generative studies

3.10.1 Pylkkänen (2008) on the addition of arguments

A well-elaborated theory of causatives is presented in the thesis “Introducing arguments” (Pylkkänen 2002) and the subsequent monograph (Pylkkänen 2008). The motivation of this study aligns closely with that of my thesis: Pylkkänen (2008, 1) aims to “focusing on the question of how arguments that are not, in a sense, “core” arguments of the verb get introduced into argument structures”. Unfortunately, the author never gives an explicit definition of core or noncore. But in the tradition in which the book is written, this term usually defines subjects and objects or single arguments of intransitive clauses. Despite similar goals, the study by Pylkkänen (2002) is implemented very differently from mine, both in the theory used and in the data analyzed. When it comes to causatives, Pylkkänen (2008) provides mainly two-participant sentences built on intransitive verbs.

Summarising their findings in Pylkkänen (2008, Tab. 3.1, p. 87), the author opposes causatives based on unaccusative verbs to causatives based on **both** unergative and transitive verbs. This division is uncommon, as most studies oppose intransitive bases to transitive ones. Such a division in Pylkkänen (2008) results from the author being primarily concerned with the nature of the argument added and not with the number of arguments of the base. Moreover, the author tends to use the notion “causative *verbs*” and draws conclusions about verbal properties, whereas the primary focus of my study and the theory I use is on whole constructions.

The main claim that Pylkkänen (2008, 83–83) makes about causatives is the following: “What universally distinguishes causative verbs from their noncausative counterparts is a syntactically implicit event argument ranging over causing events”. The postulation of any implicit constituent contradicts the principles of RRG. However, appealing in Pylkkänen’s work is the use of a well-formalized neo-Davidsonian notation for describing semantic structures. Such formalization facilitates the “translation” from one theory to another, and I use other works by Pylkkänen (namely, on Finnish, see Section 6.1.3) throughout my work.

3.10.2 Copley and Martin (2014a) on the cognitive dimension

The volume Copley and Martin (2014a) "Causation in Grammatical Structures" is a fairly recent initiative that brings together studies of the phenomenon of causation from different perspectives.

> It seems to us that linguists interested in lexical semantics study causation in lexical semantics; those interested in causation and agency study these topics in argument and event structure; and causation as it pertains to modality, aspect, and other phenomena has been addressed only in passing.
>
> Copley and Martin (2014b, 3)

Thus, the authors undertake a "quest to understand the semantics of causation via its two interfaces: the cognition–semantics interface and the syntax–semantics interface" (Copley and Martin 2014b, 2) and organize their volume accordingly. Part 1, linking cognition and semantics, is less important to my work, but it has an interesting perspective. Below, I give a detailed review of the first chapter in this part, in which the authors trace the present and outline future relationships between philosophical theories of cognition and linguistics. Part 2, about the syntax-semantics interface, is more relevant to my work. In this section, I also give a detailed review of the introductory chapter of this part. Other chapters address interesting phenomena related to the syntax of causatives, mainly about the importance of paying attention to the event structure (for instance, Lyutikova and Tatevosov (2014) and Latrouite (2014)). The reflection on the instrumental case in Hindi/Urdu causatives inspired some of my reflections in Section 6.1.3.

It needs to be mentioned that all the studies in the volume are realized within the generative approach, which is stated and justified in the Introduction:

> On the generative approach to language, theories of meaning are understood to be constrained by syntactic considerations, so there arises in the generative approach a possibility of better understanding how language represents causation by coming to see how causal meanings are syntactically structured. Copley and Martin (2014b, 3)

Even though "translating" generative studies into RRG is a challenging mental task, many ideas and findings from those works are worth it. As for the introductory chapters reviewed below, they seem to raise questions independent of the linguistic methodology one uses. In the context of the current work, the fact that these studies draw attention to specific ideas is more important than the way they do it.

3.10.2.1 Copley and Wolff (2014)

The first part of the chapter, "Theories of causation should inform linguistic theory and vice versa," presents logical or even philosophical approaches to the broadly defined phenomenon of causation. The authors suggest that theories explaining causation "by means of a dependency between the cause and the effect" (Copley and Wolff 2014, 23), although numerous, provide no understanding of important properties of causation, namely, force and energy. The authors thus present a second group of theories of causation, which they call "production theories", that characterize causation in terms of *force, energy, transmission, transference* (Copley and Wolff 2014, 24–27).

In the second part of the paper, the authors present "linguistic phenomena to which causation is relevant" (Copley and Wolff 2014, 28). These phenomena are defined from a broad semantic perspective and are reflected in multiple language constructions, such as sentence types, participant features, etc. The first phenomenon is *defeasibility*, i.e., the possibility of an accomplishment predicate not to entail the final result (like *'I built a new house, but it is still not finished'* Copley and Wolff (2014, (4), p. 31). The existence of such examples is not always taken into consideration by linguists and by philosophers working on causation. The authors' suggestion to linguists is thus to "ask whether and how our conception of causation accounts for cases of defeasibility" (Copley and Wolff 2014, 36) and to philosophers – "to pay attention to the existence of non-culminating accomplishments and their treatment in linguistic theories" (Copley and Wolff 2014, 37).

In my work, the defeasibility (also known as *implicativeness*, or referred to as *(non-) counterfactual causation*) appears to be an important feature of a construction that influences the argument marking. I offer some more discussion of it in Section 6.1.1.3

The second phenomenon addressed by the authors is about "agency and related notions" (Copley and Wolff 2014, 37). One cannot deny that this topic received huge attention, perhaps because it is the most obvious bridge (or "interface") between the event structure and the participants' realization. In this section, the authors draw attention to non-prototypical causers (cf. DeLancey (1984) and Pylkkänen (2000)), noting that *volition* might not be the right way to address the difference between types of causers. They emphasize the distinction between "teleological capability" and "direct causation" and advise that they be clearly addressed in a (cognitive linguistic) theory. The authors also offer a discussion on delimiting *volition* from *intention* and relating each of them to *preference* and *disposition*, and come up with the conclusion that *disposition* might be the most promising angle to view both prototypical and non-prototypical instances of causation (Copley and Wolff 2014, 44). It appears that theories explaining causation mainly in terms of force are most powerful in addressing dispositions and related distinctions (more details offered in Copley and Wolff (2014, 48–49)).

Throughout my work, *agentivity*, *volition*, *disposition*, and also *affectedness* occur often and in various contexts. Although I do only a limited share of semantics that directly influence the argument realization, I nevertheless try to make it conform to some type of theory that would best account for the data attested in different languages. In this respect, the formalized version of RRG that I am using is very flexible, as it allows assigning features to entities of any level.

The third phenomenon that the authors examine in connection with theories of causation is the representation of causal chains, and once again come to the conclusion that production theories fit better to explain causal chains "visible to grammar" (Copley and Wolff 2014, 54). What I find especially thoughtful is that the authors also present cases where "causal chains are not important to linguistic representations". I find it very important for both linguists and philosophers to keep in mind, so that the theories do not become overly complicated and represent not the data but the way one is thinking about it. The narrow question of causal chains is of lesser importance for my present work since I am not dealing with double causatives. Nevertheless, causatives of transitives can be regarded by some theories as causal chains if transitive verbs are supposed to be lexical causatives themselves, and I am glad to have an opportunity to claim once again that I am not using this approach for a good reason.

The whole chapter Copley and Wolff (2014), being a very thorough and insightful review itself, offers not much for the further development of my study, except for the general advice, which I was trying to keep in mind all the way.

> We have argued that certain theories of causation are more suited to certain sets of linguistic data and should therefore be utilized in developing theories of these data. This story we have been telling is thus a familiar Occam's Razor story, but the appeal to Occam's Razor is meant to be understood in a quite practical and preliminary way: we have not argued that in every case the most plausible theory of causation is the correct one. Rather, we suggest simply that one needs to be aware of the range of theories of causation while collecting, organizing, and analyzing data. Copley and Wolff (2014, 56)

3.10.2.2 Martin and Schäfer (2014)

Opening the part "From Causal Meanings to Causal Structures", the chapter by Martin and Schäfer (2014) "Causation at the syntax-semantics interface" presents the cutoff of the generative approach towards a number of issues concerning causal constructions. It is tightly linked to the opening of the first part of the same volume (reviewed right above), with section names having the most relevant terms in common. Reading these introductions one after another, one can create a clear picture of the state of the art of generative research in 2014 across the topics related to causation.

However, some traits common to most generative works make this chapter difficult to review from a theory-independent approach and even more difficult to reuse. First and foremost, it is based almost exclusively on English data, with a couple of examples from German and French. Not only does it overlook quite many relevant phenomena, but it also posits over-restricted notions like "the nominative subject" and "the accusative object" (Martin and Schäfer 2014, 210). Second, the paper pays major attention to lexical causatives, most of which do not necessarily have to be regarded as such.

The authors announce five questions they aim to answer in the paper:

> i What is the exact semantic decomposition of causative and anti-causative verbs, and how does it map to linguistically traceable event decomposition?
>
> ii At what level of linguistic representation are verbal (sub-)events encoded?
>
> iii What are the thematic as well as syntactic properties of the arguments of (anti-)causatives?
>
> iv Which verbs allow the causative alternation, and why do verbs which are best characterized as causatives not all have an anticausative counterpart?
>
> v Finally, what is the derivational relationship between the alternants, i.e., is the causative version derived from the anti-causative version via a process of causativization, or is the anti-causative version derived via a process of detransitivization; alternatively, are both derived from a common base? Martin and Schäfer (2014, 209–210)

Unfortunately, the paper ends abruptly, without offering a summary of the authors' answers to these questions. It might not be, however, the goal of the chapter: in contrast to Copley and Wolff (2014), aiming to tell *what* to look at, Martin and Schäfer (2014) tells *how* to look.

Question (iii) is the most relevant to my work. It is claimed to be addressed in Section 4 of Martin and Schäfer (2014), but in fact this section discusses only causers, asking whether they must be agents and, if not, to what extent.

To conclude this rather brief review, I just want to mention that the chapter is a typical representative of generative works about causatives and can be used as a summary of prior works of that kind, but, like most works of that kind, it does not bring much to my work.

3.10.3 Hoffman (1991) on accusative vs. oblique causee languages

The dissertation, "The syntax of argument-structure-changing morphology," adopts a generative approach to causatives and applicatives. In terms of theory, it relies mainly on Marantz (1984), Marantz (1985), Rosen (1990), Grimshaw (1990), and Li (1990) as well as on "class notes" by Chomsky.

What makes it especially relevant for my work is its focus on three-participant constructions (relatively rare compared to a massive corpus of studies where "causative" is implicitly synonymous with "causative of intransitive"). Such an ambitious title, however, appears to tackle a particular question (which is probably quite a distinctive property of dissertation titles).

The main theoretical concept explored throughout the dissertation is "case licensing". Put simply, in less theory-specific terms, throughout Hoffman (1991), it is the ability of a predicate to take an object (which, according to the author, differs from other analyses in generative studies, namely, Li (1990)). As summarised in Hoffman (1991, 35–37), a predicate is assigned the feature [±L], which tells whether it "can activate an LP for licensing". Under LP, the author understands what is called ARG_O by Chomsky, which in his works stands for a unit bearing the Accusative case. Thus, "can activate an LP for licensing" roughly corresponds to the ability to have an accusative object.

Working with synthetic (in terms of Shibatani (1976b), *built using overt morphology*) causatives, the author suggests that the causative morpheme is a separate predicate taking "the base verb as a complement" (Hoffman 1991, 154). The cross-linguistic variation of possible argument marking thus results from the values of the [L] feature on the causative predicate and the base verb. The typology resulting from this analysis could be surprising:

> One way to characterize the differences among causatives across languages is to divide them between "accusative causee" languages, in which the causee is marked as a normal direct object, and the object of the base verb may be either oblique or also accusative, and "oblique causee" languages, in which the object of the base verb retains its direct object marking, and the causee is expressed obliquely. Hoffman (1991, 158)

The question that seems to interest the author the most is "Why in some languages the causee can be accusative, and in some others – cannot?" The very question seems to be strongly influenced by the data from Bantu languages, although the author lists as examples of "accusative causee languages" also "Chamorro [(Austronesian)] and some dialects of Quechua" Hoffman (1991, 159). The author comes up with three options yielded by different combinations of the values of [L] feature at the causative predicate and the base verb:

1. only the object of the base verb "is L-licensed", i. e., can be object,
2. only the causee can be the object,
3. both the causee and the base object are L-licensed.

Interestingly, the author defines only two types, merging 2 and 3 from the above list. This decision again underscores the author's primary interest in accusative causees, and other data seem to serve merely as background for this problem. (And although the author provides data showing that a single language can have constructions of different types, I cannot help but notice the whole reasoning going in terms of "languages". It does not change the conclusions, though, and might be due to the optimism of linguistics some decades ago in offering various classifications of languages.)

The causer explains this variety by a lexical property of a causative affix (Hoffman 1991, 156). Thus, languages (rather constructions) with oblique (not object-like) causees are due to the value [-L] of the causative predicate. In asymmetrical constructions where only causees are true objects, the causative affix is [+L], and moreover, it has the "lexical property of suppressing [+L] on the lower verbs" (Hoffman 1991, 175). Finally, symmetrical constructions are built with the use of a causative morpheme that has [+L] itself and does not suppress [+L] on the lower verb.

To put it as briefly as possible and free from special terminology, the central claim of the Hoffman (1991) is that the causative morpheme is exclusively responsible for encoding the causee as object or not, and also for a language having double objects in causatives or not. Although the syntactic trees in my work look very different, I can still make this claim, or at least bear it in mind, while analyzing my data. Throughout the Part II, I try to find factors influencing the shape of causative constructions and determine whether they apply to all causatives in a language (i. e., can be regarded as a property of the causative morpheme) or to some lexemes only, or are applicable at a different level.

Part II

Theoretical part

Chapter 4

Introduction

The primary aim of this thesis is to provide an RRG analysis of three-argument constructions featuring a morphological causative. Given that these constructions have previously been explored across various languages using different theoretical frameworks, a substantial part of my analysis involves "translating" earlier findings into RRG. While this task may seem somewhat repetitive, it offers several advantages. Primarily, if an analysis applied to a specific language can be effectively translated into RRG, it serves to validate its well-formedness and cross-linguistic comparability. Consequently, my work assesses the contributions of prior studies even though it is framed within a particular theoretical approach. Moreover, RRG's ability to integrate multiple previously established analyses of individual languages underscores its theoretical rigor and its potential to encompass a broader range of language phenomena.

Part II following this introduction is organized as detailed below. Chapter 5 provides an extensive background for my thesis and outlines the methods I employ throughout my research. My reflections begin with a typology of causative constructions primarily grounded in Dixon (2000), which has significantly shaped and informed my research. In Section 5.1, I offer a detailed critique of this paper, summarising my criticisms in Section 5.2. The languages mentioned in Dixon (2000) largely influenced my selection of languages. Initially, I aimed to challenge some of the choices made by Dixon (2000) regarding specific languages. Later, I expanded my sample to include additional languages to ensure a broader representation.

In Section 5.3, I present the background for the methodology I utilize throughout my analysis. The central idea in Kemmer and Verhagen (1994) and in my analysis presented in Chapter 6 is to connect causative constructions with other constructions within the same language. Unlike other approaches, the "relatable" constructions in my analysis are not the sources from which causatives are derived (i. e., not just simple transitive or intransitive constructions). Instead, the focus is on the morphosyntactic marking of arguments in two somewhat similar constructions, i.e., that can be related. While this approach may appear overly syntactic, it reveals important syntactic and semantic similarities between different constructions in each language. Insights drawn from individual languages form a basis for cross-linguistic generalizations.

This paper has already been reviewed in Section 3.7 with an emphasis on positioning this research within the existing literature. In contrast, Section 5.3 delves into the specifics of the method that links causative constructions to others. Building on the general methodology proposed in Kemmer and Verhagen (1994), I extend it to yield more precise analyses of structurally diverse languages. A comprehensive summary of my approach is presented in Section 5.4.

Chapter 6 presents my examination of causative strategies across various languages. The goal is to compile a comprehensive enumeration of all the "relatable constructions", i.e., constructions showing argument patterns that resemble those of causative-of-transitive constructions. The chapter opens by discussing the argument-marking strategies that have received the most attention in previous scholarship and then addresses less commonly examined strategies. Each section of Chapter 6 presents data from one or more languages, along with previous analyses. In presenting analyses proposed by other theoretical frameworks, I strive to align them more closely with RRG. Subsequently, I provide RRG analyses of the languages in question, determining the status of arguments and their macroroles. To answer these questions, I also analyze other constructions within the same language (e.g., ditransitives, passives, and others) to identify the functions of the morphological markers used in causative constructions. This reasoning often leads to re-evaluating Dixon's type (Dixon 2000) of causative construction available in each language.

Generalizations based on the analyses of individual languages are summarised in Chapter 7. First, all findings related to the languages are itemized. Tab. 24 outlines the causative strategies identified in each language within the sample. These observations are further generalized to create constructional schemas—structures that encapsulate various types of information found in each construction. The schemas are listed in Tab. 25.

In summary, Part II encompasses both the methodological and theoretical frameworks of my thesis, providing a comprehensive resource for those interested in typology and Role and Reference Grammar. The subsequent computational analysis is presented in later sections, further expanding on the insights gained from this theoretical foundation.

Chapter 5

Background and method

This chapter is crucial for understanding the overall thesis because it explains the main methodological principles that guide the research presented herein.

5.1 Dixon (2000): a starting point

As Chapter 3 shows, the argument structure of causative constructions has not been studied much from a typological perspective. In this section, I would like to discuss the study by Dixon (2000), which proposes a typology of five varieties of constructions with morphological causatives based on transitive verbs. This study appears to have several inconsistencies, which I will try to make visible throughout the present chapter.

type	causer	original A (causee)	original O
(i)	A	special marking	O
(ii)	A	retains A-marking	O
(iii)	A	has O-marking	has O-marking
(iv)	A	O	non-core
(v)	A	non-core	O

Table 9: Causative of transitive types (reproduced from Dixon 2000, p. 48, (24))

Dixon (2000) suggests five types of marking patterns in constructions with morphological causatives based on transitive verbs, see Tab. 9. The core idea of the approach is to trace how the marking of the three arguments of a causative construction is related to argument marking in a non-causative transitive construction from which it is derived. The label *A* stands for the agent (in this context, equal to the syntactic subject), and the label *O* stands for the direct object. Only these two participants are considered *core* arguments in Dixon's terms. These notions are combined to define participants as a whole, including their syntactic behavior and morphological marking. Sometimes, however, the author refers only to morphological properties: cf. "O" = 'is a direct object' vs. "has O-marking" = no claim about the syntactic role.

The general idea of relating causative constructions to sentences from which they are derived seems natural if one regards causatives as derivations. The derivational approach assumes that causative constructions take basic transitive patterns and somehow modify them. As a result, the analysis inspired by the derivational approach tells how exactly causative constructions are derived, but pays secondary attention to what exactly is created after the derivation.

In contrast to this approach, I would like to create a careful enumeration of the existing variants of argument structure in causatives-of-transitives as the first step for further generalizations. As my data show, they are more numerous than the types listed by Dixon (2000). In the remainder of this section, I present Dixon's classification type by Type and discuss its limitations.

5.1.1 Type (i)

Type (i) listed in Tab. 9 is defined as follows: "In type (i) there is a special marking, used just for the causee in a causative construction" (Dixon 2000, 48). To exemplify this strategy, Dixon (2000, 48) cites Nivkh (isolate, Sakhalin) and Telugu (Dravidian, India). Nivkh is included in my sample and discussed in detail in Section 6.1.4. As for Telugu, the marker *cēta* is used not only in causative constructions but also in passives (see examples below). Hence, Telugu has to be classified differently.

(21) nēnu rāmu-cēta kāru-∅ kaḍig-inc-ǣ-nu — TELUGU
I Ramu-INS car-ACC wash-CAUS-PST-1SG
'I had the car washed by Ramu'/ 'I caused the car to be washed by Ramu.'
Krishnamurti 2003, p. 433, (20f)

(22) rāmu rawḍīl-a-cēta koṭṭ-a-baḍ-ḍ-āḍu — TELUGU
Ramu rowdies-OBL-INS beat-INF-suffer-PST-3M-SG
'Ramu was beaten by rowdies/hooligans.' Krishnamurti 2003, p. 433, (20h)

At first glance, the strategy of using a special marker for a single construction appears quite costly for a language. If a language has some types of three-argument constructions, what makes it create a new one for causative constructions only? If a language lacks three-participant constructions, it would definitely reuse devices from two-argument ones for causatives-of-transitives.

In fact, Dixon himself remains undecided on what counts as a "special marker" to put a language under Type (i) and what does not. Consider his comment about Sinhalese (Indo-European, Sri Lanka): "We find a variant of this pattern in Sinhalese where the original A of a simple transitive or ditransitive verb goes into the dative case, but this must be followed by the postposition *kiəla* (Gair 1970: 68–70). **(We could regard this as a language of type (i), with dative-plus-kiəla constituting 'special mark-**

ing' of original A)" (Dixon (2000, 55), emphasis mine – VG). Nevertheless, Sinhalese is classified as Type (iv) in Dixon (2000, 55).

My approach pushes to the extreme the principle of finding similarities between different constructions (see details on the methodology I use in Section 5.4. In this line, Type (i) should be regarded with major caution. I suggest (see Chapter 6) assigning this Type to languages only as a fallback strategy in case of insufficient language data.

5.1.2 Type (ii)

Type (ii) is a type where "both causer and original A receive A-marking" (Dixon 2000, 48). Two languages are listed as examples: Trumai and Kabardian. (Qiang is also mentioned, but I did not include it in my current sample, leaving these data for future research.) According to this definition, Type (ii) appears to introduce a *referential ambiguity* (Nieuwland and Van Berkum 2008) also known as *referential conflict* (Kibrik 1987; Kibrik 1991): a situation, in which "readers or listeners are unable to select a unique referent for a linguistic expression out of multiple candidates" (Nieuwland and Van Berkum 2008, 603). Dixon (2000, 49) reports that the referential conflict in constructions of Type (ii) is avoided due to strict word order in such sentences. One factor leading to a referential conflict could be the lack of morphological means to express three arguments in three distinct ways. Hence, the word order serves as an additional disambiguation device.

Both languages listed by Dixon (2000, 48) as Type (ii) receive due attention in Chapter 6. Kabardian (Section 6.2.2.2), in fact, appears to have only two cases eligible for marking core (in RRG terms) arguments. In addition, it cross-references arguments on the verb, which must not be neglected when studying the argument structure of any sentence. As for Trumai (Section 6.2.2.3), its basic alignment is unclear; hence, identifying the agent is also not straightforward.

5.1.3 Type (iii)

Type (iii) is defined as follows: "original A takes on O-marking, original O retains O-marking" (Dixon 2000, 50). It is similar to Type (ii) in that respect that it also contains two identically marked arguments.

Dixon (2000, 51–52) provides many example languages having causatives of Type (iii). However, this group of languages is not uniform. Some of them – according to Dixon and Aikhenvald (2000a, 50), "most of these languages" – have two arguments with identical morphological encoding but different syntactic properties. Since Bresnan and Moshi (1990), these constructions are often referred to as *asymmetrical.* The second group of Type (iii) languages has two "full objects", i.e., two arguments with an identical number of syntactic object-like properties. These languages and constructions are often referred to as *symmetrical*.

Dixon (2000) himself does not introduce any special subtypes for Type (iii) but discusses the difference between asymmetrical and symmetrical constructions (in different terms, though). Since in asymmetrical languages, the two arguments bearing the object-like morphology are not identical, Dixon (2000, 50) suggests that they might be regarded as Type (iv): "[i]t is generally the original A which is the full O while the original O has become the second object **(making these ostensively type (iii) languages perhaps a special case of type (iv), where A becomes O and the original O moves out of the core)**" (emphasis mine – VG). In other words, asymmetrical Type (iii) constructions demonstrate a step-wise demotion: the agent of the base verb becomes the (full) object of a causative verb, and the object of the base verb moves to some other, further demoted position. The peculiarity of Type (iii) is that this position requires the same morphological coding as the position of the (full) direct object.

Although the general idea of tracing the mechanisms of demotion seems fruitful, this note once again questions the criteria used by Dixon (2000) for distinguishing the types from each other and for classifying individual languages. The status of the 'second object' position is unclear, both in individual languages and in linguistic analyses. How far is an object demoted? How close does this construction get to Type (iv)? Is this second object still an argument at all?

In Chapter 6, I engage with three varieties of Type (iii) – in Dixon's wide sense – languages. In Section 6.2.1.2, I discuss symmetrical and asymmetrical causative constructions in Bantu languages, as well as the co-existence of both types of constructions within a single language. In addition, in Section 6.2.3, I explore causative constructions with object-like marked arguments in languages that allow differential object marking and (quasi)incorporation of one of the participants into the verbal complex.

5.1.4 Type (iv)

Type (iv) suggests that the "original A becomes new O, original O moves out of the core" (Dixon 2000, 52). Hence, the resulting construction has two (instead of three) *core arguments* in Dixon's terms.

The RRG's understanding of the core arguments differs from Dixon's. For Dixon and Aikhenvald (2000b, 2–3), only the subject and the direct object are core arguments. As for the recipient, it is considered an extension to core (Dixon and Aikhenvald 2000b, 3). In these terms, no three-argument core is possible without the doubling of some position. As for RRG, it treats all arguments that are semantically obligatory for a verb as CORE (see details in Section 2.4, Section 2.5.1). RRG analyses allow for cores of more than two arguments. Thus, Dixon's Type (iv) constructions can still be of interest for the present

RRG study of three-argument causative constructions.

Additionally, the definition above seems not quite useful, as moving out of the core does not entail any specific marking. There are plenty of non-core (i.e., non-subject and non-direct object) positions available for a constituent. So, this Type would accumulate very different structures. It seems to have been a better solution to split this group of languages into several types in the updated typology, based on how exactly the constituents are marked and what other constructions they reflect.

As examples of Type (iv), Dixon (2000, 52–53) mentions a handful of languages, including some Bantu languages (namely, Swahili and Babungo) and some Austronesian languages (including Javanese). The latter is given special attention in Section 6.1.2.3.

5.1.5 Type (v)

Type (v) suggests that "original O stays as O, original A moves out of the core" (Dixon 2000, 54). Under this Type, Dixon (2000) explicitly suggests two subtypes. The first subtype (v-a) describes constructions where "[m]arking of original A is motivated by a hierarchy" (Dixon 2000, 54). Under the term *hierarchy*, Comrie's paradigm case (Comrie 1974) is understood. It predicts that an argument would occupy the first (leftmost) available slot following the hierarchy (23).

(23) SBJ → DO → IO → oblique → GEN → object of comparison

Dixon notes that this pattern is not as default and paradigmatic as believed (compare also with Section 3.3).

> In fact, this pattern is rather rare. It is found in Western Romance languages such as French and Italian. Comrie quotes Turkish as a further example, but the literature on this language gives mixed information. <...> There may be a few other languages of Type (v-a), but they are greatly outnumbered by those of Type (v-b). When we also take into account types (i–iv), it will be seen that there is no justification for attaching special importance to the pattern illustrated for French. Dixon (2000, 54–55)

What is of major importance for the method of the present study is that the paradigm case is more oriented towards explaining the process of verbal derivation rather than its result. In other words, Comrie's hierarchy does not describe any construction as a single pattern contrasted with other patterns. What it does is describe how causative constructions with different numbers of arguments behave in a language. So, the motivation for why a certain pattern is attested in a given construction seems not to be a relevant classification principle for the enumeration of argument-marking strategies I aim for in Chapter 6. For causatives-of-transitives, the paradigm case predicts the "landing" of the causee in the indirect object position, but the reality appears to be much more complex.

Therefore, I totally discard the idea of explaining my data with the paradigm case, even though I agree that this trend is attested in many languages. Instead, I would like to study whether the causee has the same morphological and syntactic properties as the recipient-like object in a ditransitive construction.

The second subtype of patterns, (v-b), is defined as the one where the "[o]riginal A is assigned a fixed non-core function (irrespective of whether the underlying clause is simple transitive or ditransitive)" (Dixon 2000, 55). Dixon (2000, 55–56) lists six morphological cases that can be used in this position: DAT, INS, LOC, ALL, ADESS, POSS:ESSIVE.

This list raises questions. First, this list is far from being exhaustive. For example, Bashkir has a strategy with ABL, which is not listed. It might be that in other languages, there are more marking strategies also overlooked by this list. Second, the question of delimiting Types (v-a) and (v-b) emerges. If the slot of an indirect object on Comrie's hierarchy requires dative marking in a given language, how could one tell whether it is (v-a), i.e., following the paradigm case, or (v-b), i. e., requiring a fixed DAT? Even though this question has been answered, what would its added value be for the linguistic analysis?

These gaps inform the analysis method I use, which relates all causatives to other constructions (not necessarily those from which they are derived), thereby employing a similar argument encoding. This approach would create a list of functional positions to which arguments of a causative construction can be similar. Moreover, it brings insightful generalizations. Namely, Finnish, which is cited as the single language having ADESS causees (Dixon 2000, 55), happens to be relatable to instrumental constructions and thus takes place in a wider group of languages (see details in Section 6.1.3).

To sum up, Dixon (2000) is a good starting point for creating an enumeration of existing argument-marking strategies in causatives of transitives. It cannot be used as is due to several drawbacks. First, the criteria for distinguishing the types are unclear and yield ambiguous results (even within the chapter itself). Second, the typology provides little precision about the results of causative derivation, focusing mainly on the mechanism of deriving causatives from transitives. Third, Dixon's typology lacks a functional dimension and can barely serve comparative purposes. The next sections will discuss how these inconveniences can be overcome to provide a better account of the strategies in question.

5.2 Criticism of Dixon (2000) summarized

As noted in the previous section, Dixon's typology is an effective launching pad, as it lists various options for encoding all participants. However, like many studies of causatives, it still provides some "dynamic" reasoning, describing not only the result of the derivation but also the process by which it occurs. This is, for instance, reflected in the terminology "retains", "receives", "becomes". In my work, I would like to focus more on the resulting construction, leaving the process of derivation out of focus.

The second main difference between my focus and Dixon's is the definition of positions in functional terms. When Dixon's typology gives precise case names, it does not relate them to anything. In contrast, instead of naming the cases (which can still be challenging in some languages), I would like to refer to them by identifying their functions in a language. Thus, I prefer the term "recipient case" to "dative" (since languages in which the recipient case is not glossed as dative exist). The difference becomes even more important when separating "the case used for prototypical instruments" and "the case used for demoted agents in passive clauses", since languages demonstrate different patterns of syncretic marking related to these positions. One could argue that if a case has already been glossed as "instrumental," it has already been observed as one marking prototypical instruments. In reality, glosses most often reflect a way of interpreting the data rather than the data itself, along with the tradition followed by each researcher, and thus gloss labels should not always be trusted as final or unquestionable definitions.

In this respect, I am very careful to formulate my observations as comparative concepts, as they are introduced in Haspelmath (2010). Consider the definition of 'dative' as a comparative concept formulated in Haspelmath (2010, 666, (2)):

> A dative case is a morphological marker that has among its functions the coding of the recipient argument of a physical transfer verb (such as 'give', 'lend', 'sell', 'hand') when this is coded differently from the theme argument.

The discussion following this definition specifically refers to causative constructions.

> This definition is based on the conceptual-semantic concepts 'recipient' and 'physical transfer verb', as well as the comparative concepts 'morphological' and 'argument'. The comparative concept 'morphological', in turn, presupposes the comparative concept 'word', perhaps definable as a 'segment string that cannot be interrupted by a free form without changing the meaning', and the comparative concept 'argument' can be defined as 'referential phrase that can fill a verb's semantic valency position'.
>
> <...>
>
> The Russian Dative matches the 'dative case' concept in (**??**), and so does the Finnish Allative, but the Nivkh Dative-Accusative case, for instance, does not match it, despite its name, as it is used to express the causee in causative constructions, but not the recipient of a physical transfer verb. Note that we cannot say that the Russian Dative and the Finnish Allative 'instantiate' the 'dative case' concept because these categories have many

> more properties than are contained in the definition in (??). This is the crucial difference between comparative concepts as proposed here and the crosslinguistic categories that I reject. On the categorial-universalist view, categories in particular languages instantiate crosslinguistic categories, and they can thus be equated across languages. On the view presented here, the fact that two language-particular categories both match a comparative concept just means that they are similar in the relevant respect, but not that they are 'the same' in any sense. Haspelmath (2010, 666)

In the present work, I have no aim of formulating precise comparative concepts related to causative constructions, nor do I claim that the concepts I use can serve as comparative for a typologically representative sample. Nevertheless, I try to avoid naming a language-specific category without at least some comparative explanation.

To summarize, I take Dixon's five types as a preliminary sketch of possible argument-marking strategies and build my own enumeration, offering more precise details and supporting my decisions with more semantically motivated reasoning grounded in comparative concepts.

In the next section, I give more details on the mechanism of comparison I use, inspired by Kemmer and Verhagen (1994).

5.3 Inspiration from Kemmer and Verhagen (1994)

The paper that has conditioned my choice of general methodology – a theory-independent, semantically and functionally motivated analysis – is Kemmer and Verhagen (1994). I have already offered the general review of this approach in Section 3.7. In this section, I would like to present the subtle details of their analysis and explain the expansion I decided to use throughout my work.

Before addressing the analysis in Kemmer and Verhagen (1994), I need to make some notes on the linguistic material explored by the authors. First, they claim to focus mainly on analytical causatives. However, they suggest that the same line of analysis should hold for morphological causatives as well, since "in that case, the causal and effected predicate happen to be expressed in one word composed of two morphemes, one indicating the causal predicate and the other the effected predicate" (Kemmer and Verhagen 1994, 118). Therefore, I found it appropriate to ground my analysis of morphological causatives on the same premises that were explored in Kemmer and Verhagen (1994). Second, the authors suggest that the variety of argument marking in causatives-of-transitives is limited exclusively to the variety of the causee-marking, the causer, and the affectee being always marked in the same way (as arguments of a simple transitive clause). However, Dixon (2000) provides examples of Type (iv) languages where the causee is marked like the object of a simple transitive, and the affectee receives a different marking. In my analysis, I expand the general principles of analysis offered in Kemmer and Verhagen (1994) to include the encoding of the affectee as well, even though languages of Dixon's type (iv) are only marginally represented in my sample.

The key notion of the analysis proposed in Kemmer and Verhagen (1994) is the notion of *structural template*. It is not entirely clear what is included in this notion, but from its definition (see below) and the limitations the authors address, this template seems to encompass morphological and syntactic properties of participants, but not necessarily their semantic roles.

> First of all, we propose that ICs [causatives of intransitives] show subject and direct object marking on their principal participants, just like a simple transitive clause, because they take the simple transitive clause as a **structural and conceptual model** (emphasis mine – VG). In other words, clauses of the pattern *I ate the cake* are the pattern for clauses of the type *I made Terry cry*.
>
> Secondly, the case marking patterns in TCs [causatives of transitives] noted by Comrie are of two basic types, parallel to two kinds of simple clauses. In some languages, the TC structure is modelled on the structure of a ditransitive clause (subject, object, and indirect object marking on the three participants). Thus, clauses of the type *I gave her the apple* are used in some languages as a basic template for more complex causative structures of the type *I made John do it*. In other languages, for example Finnish, the model for TCs is provided by sentences of the type: *I hit it with a hammer* <...> Some languages allow for more than one possibility <...>.
>
> Kemmer and Verhagen (1994, 125)

The definition above lists two possibilities for causee encoding, which might be the two most commonly used in the languages of the world (although a thorough quantitative study to verify this hypothesis is needed). The similarity of causatives and ditransitives seems to be supported by many types of data (including the grammaticalization of causative affixes from the verb 'give' in some languages, which is also noted in Kemmer and Verhagen (1994, 129)). Compare the claim in Dixon (2000, 52): "The causative of a transitive is a kind of ditransitive clause. In many languages, it has essentially the same syntax as a non-causative ditransitive (involving a verb like 'give' or 'show' or 'tell')". To represent the relation between simple clauses and causatives, Kemmer and Verhagen (1994, 126) offers a table that captures the participant-to-participant relation. In my work, I would suggest using indexed formulae in order to visualize the same type of data. Thus, the fact that causees are often realized as recipients can be represented as follows:

(24) $E_i^{caus}\,C_j^{caus}\,A_k^{caus}\;:\;A_i^{ditr}\,R_j^{ditr}\,T_k^{ditr}$

The capital letters stand for the participants of each of the construction (**E**ffector – to have a different initial than **C**ausee, **A**ffectee; **A**gent, **R**ecipient, **T**heme); the upper index tells the type of construction these participants appear in (in this formula, **caus**ative and **ditr**ansitive); the subscript letter indices the identity of marking (in this formula, pairwise: effector = agent = *i*, causee = recipient = *j*, affectee = theme = *k*).

The salience of the relationship between causatives and ditransitives is not questioned. Nevertheless, the reference ditransitive construction must be selected with great caution. As highlighted in Margetts and Austin (2007) (after Comrie (2003), Newman (1998), and Margetts (2007)), the verb 'give' is often irregular and thus is not a typical representative of a ditransitive class, if there is any. In other words, if one attests a similarity between the argument pattern of 'give' and the argument pattern of a causative, they both likely exhibit the same ditransitive-like pattern. However, if the causative pattern is not identical to the pattern for 'give', it does not mean that the causative does not belong to the ditransitive class – perhaps, it is the verb 'give' that does not. In my research, I try to account for this situation and, in some languages, I come across the peculiarities of the 'give' verbs rather than those of causatives. For a quantitative typological study, a strict formal definition of the ditransitive class would be required.

The second relation (causee = instrument), although it seems intuitive, appears to entail some questions and difficulties. Already while discussing the very first French examples, Kemmer and Verhagen (1994, 130) notice that French "distinguishes marking of simple physical Instruments from marking of more abstract means", consider (25).

(25) a.

Creus-er	le	sol	avec	une	pioche	FRENCH
dig-INF	DET.DEF.M.SG	ground	with₁	DET.INDF.F.SG	pickaxe△	

'To dig the ground with a pickaxe.'□

Larousse.fr. Dictionnaire de français Accessed: 2026-05-01

b.

Elle	a	prouv-é	ça	par	des	exemple-s	FRENCH
she	AUX.3SG	prove-PTCP.PST	this	with₂	DET.INDF.PL	example-PL△	

'She has proven that by means of examples.' Kemmer and Verhagen 1994, p. 130

Importantly, only *par* (glossed as 'with$_2$' above) can introduce causees in French causatives, but never *avec* (glossed as 'with$_2$'). For this particular case of French, the authors insist on relating *par*-causees to instruments: "We consider such means-instrument constructions to fall under the rubric of an instrumental clause type and expect that the semantics of the specific instrumental marker in a given language will figure in the relation of the simple and the TC causative clause types" (Kemmer and Verhagen 1994, 130).

However, when coming to German examples (Kemmer and Verhagen 1994, 132), the authors attest an alternation between two different cases: accusative and what they call "agentive" (introduced by the preposition *von*).

(26) a. Er liess sein-en Sohn den Brief abtippen — GERMAN
he let his-ACC son DEF.M.SG.ACC letter offtype△
'He made his son type the letter.'

b. Er liess den Brief von sein-em Sohn abtippen
he let DEF.M.SG.ACC letter by his-DAT son offtype△
'He had the letter typed by his son.' Kemmer and Verhagen (1994, p. 132, (7))

None of these cases was predicted in the initial statement of possible relatable constructions. The authors explain the difference between two German causative constructions with the same criteria as they explain the competition of constructions with dative and instrumental causees in Hindi and Kannada (Kemmer and Verhagen 1994, 131, 132). As a result, the authors suggest an (intermediate) conclusion that posits an opposition "agentive or instrumental vs. dative or accusative", shifting on the fly to a tripartite distinction "accusative vs. dative vs. instrumental or agentive". Consider a larger context below.

> It [the peripherality of causee coding] also relates to which participant is viewed as more affected: the more affected the participant, the more object-like and, therefore, central it is to the event as a whole. Thus, agentive and instrumental marking (which may, in some languages, be the same marking) are consistently associated with more peripheral participants, which have more control over their actions and are less directly affected. In contrast, dative and accusative participants are more central participants with less independence of action and a greater degree of conceived affectedness. The dative is intermediate in terms of these properties, while the accusative is the polar opposite of the instrumental and/or agentive.
>
> Kemmer and Verhagen (1994, 133–134)

Language	**Possibilities for causee marking**	**Source(s)**
Dutch	∅ vs. Dative vs. Agentive/instrumental	Kemmer and Verhagen (1994)
French	Dative vs. Agentive/instrumental	Hyman and Zimmer (1976)
German	Accusative vs. Agentive/instrumental	Comrie (1976)
Hindi	Dative vs. Instrumental	Saksena (1980)
Kannada	Dative vs. Instrumental	Cole (1983)
Mongolian	Dative (= Agentive) vs. Instrumental	(not given)
Quechua	Accusative vs. Dative vs. Instrumental	Cole (1983)

Table 10: Variation in case marking on causee of causative-of-transitive (reproduced with minor modifications from Kemmer and Verhagen 1994, 137)

At the end of the same section, the authors provide a list of languages with several causative-of-transitive constructions and annotate the cases used for causee marking (see Tab. 10). What is most important for the current focus of this section is the partial overlapping of instrumental and agentive in some languages. These data seem to need further specification. Thus, if the French *par* (glossed as 'with2' in (25)) can be regarded as both agentive and instrumental, or, to be more precise, a syncretic marker for both agentive and instrumental, the German *von* in (26) definitely can not. As for Kannada, according to Dryer (1982, 313), instrumental is also used for agents in passive constructions; thus, it should be labeled as "Agentive/instrumental". The syncretism of dative and agentive in Mongolian is unknown to me. Argument strategies used in its not quite close relatives, namely, Kalmyk (Mongolic) and also Bashkir and Mishar Tatar (Turkic), are in fact peculiar, see Section 6.1.1.4).

Thus, two important messages follow from (25). First, there are other constructions in addition to ditransitive and instrumental to which causatives-of-transitives can be related. Second, the instrumental case is often, but not always, syncretic with the "agentive" case, i. e., the one used for demoted agents in passive constructions.

Relating causatives to instrumental clauses seems quite logical from both syntactic and semantic points of view. The semantic reasoning provided in Kemmer and Verhagen (1994) relies mainly on relating the causee to some kind of an animate instrument, although not a prototypical one. Indeed, in many causative constructions (especially with non-defeasible ones), the instrumental component of the meaning is very prominent. From the syntactic point of view, in RRG, instruments can be oblique core arguments, while demoted agents in passives usually are not. The default rule of assigning the instrumental case in RRG is, very basically, to the second-agentive participant of a clause (which is the causee, given the choice of three options), see details in Section 2.5.3. Thus, treating cases of agentive/instrumental syncretism as instrumental has its advantages.

In my sample, there are three types of languages related to instrumental/agentive marking. First, languages like Finnish (Section 6.1.3), where causees are marked like instruments but differently from demoted agents in passives. Second, languages like Bashkir (Section 6.1.1.4), where causees are marked like demoted agents, but unlike instruments. Among the languages listed in Tab. 10, German *lassen*-causatives belong to this type. There is no reason to postulate any underlying passivization in these constructions, like Comrie (1976) did. The same reasoning of just relating the participants and finding similarities in the usage of case markers applies to passive constructions as it applies to ditransitive and instrumental ones. Third, languages like Kannada (Section 6.1.1.1.3), where the same marker is used for all three types of participants: causees, instruments, and demoted agents. For clear cases, like Finnish and Bashkir, I claim a relation to a single construction in my analysis. Since disambiguation between instrumental and agentive is not systematically done in most studies, I tried to find data to support the relation to a single construction where possible. Though there are still languages that genuinely have a syncretic marker for both demoted agents and prototypical instruments, and the same marker is used for causees. Contrary to the decision made by Kemmer and Verhagen (1994), I decided to put these syncretic languages under the same type as Bashkir, i. e., to highlight their relation to demoted agents rather than to instruments. The main motivation for doing so was that in quite many other studies (including Kemmer and Verhagen (1994); also Van Valin (2018a) and some others), the same analysis is applied to Kannada (which is the language of a syncretic type) and to French (which can be related to the instrumental type with a certain limitation, but undoubtedly fits the demoted agent type).

While discussing the fact that causee marking sometimes resembles the marking of demoted agents in passive constructions, Kemmer and Verhagen (1994) find it unsurprising for their semantic approach and the opposition between more topical (dative) and less topical (instrumental) causees: "passive agents are also rather nontopical, fairly dispensable participants. If the agent *were* (authors' emphasis) topical, of course, one would not use a passive construction at all" (Kemmer and Verhagen 1994, 135). The authors emphasize that Finnish has causees marked as instruments, but not demoted agents, which speaks against analyzing all syncretic cases as relatable to demoted agents. Although the authors seem to lack an example of a language where causees are relatable to demoted agents, not instruments (meanwhile, German could serve as a good example), they offer a possible explanation for this situation: "Where the marking found on causees and passive agents excludes (prototypical) instruments, then the marker incorporates the just-mentioned specification in its semantics, but in addition specifies agentivity" (Kemmer and Verhagen 1994, 136). The "just mentioned specification" is "a dispensible, nontopical causal intermediary", and it arises with instrumental causees. Thus, the authors seem to stick to the instrumental analysis of syncretic forms (and even of German) in the absence of more data. Since I have the data from Bashkir, I can offer a more precise analysis of all three types of causees.

Kemmer and Verhagen (1994) is by far not the only study that uses the reasoning of comparing two constructions of a language to learn more about one of them. Internal language comparison has been a standard in typological studies for some time. For example, WALS, one of the most renowned typological databases, has a chapter and a map that answer the question: "Whether comitatives are marked in the same way as (prototypical) instruments?" (Stolz, Stroh, and Urdze 2013) One could call this an investigation of case syncretism (which, *à propos*, received quite some attention within RRG in works by Prof. Nakamura, Nakamura (2011) and Nakamura (2021)), which is also a fair definition of the problem. From a methodological point of view, this chapter relates one construction (comitative) to another (instrumental) and identifies the similarities and differences in the markers used. To cite more examples, the recently released Grambank (Skirgård et al. 2023) formulates almost all titles of its maps as binary questions, among which one can find, for instance, the following: "Can the recipient in a ditransitive construction be marked like the monotransitive patient?" (Feature GB105). This relates one type of argument marking in ditransitive constructions to another construction, namely, the monotransitive construction. These and many other studies support the usage of the method presented in Kemmer and Verhagen (1994) for other research questions and other materials.

5.4 My approach illustrated

In this last section, before presenting my analysis (see Chapter 6), I would like to offer a teaser of how it goes and what one might wish to pay attention to while reading[9].

First of all, the outcome of my analysis is not a typology in its proper sense. To identify the type of work done for this chapter and the data structure that resulted from it, I would rather use the word *enumeration*, i.e., an exhaustive list of possible options. In other words, my goal was to present all the varieties of how argument marking in a construction with a three-argument morphological causative might look like – and put them as entities of the enumeration. The crucial difference from a typology is that I do not assign weights to these entities, considering them equally likely from a theoretical point of view. A later quantitative typological study would show how many languages fall into each category and how they are distributed worldwide.

The languages that are discussed in this chapter thus have not been selected on the premises of typological representativeness. Instead, I focused mainly on languages that have already received attention in papers discussing causative constructions. This allows me to make my results comparable to the prior scholarship. For some options in the enumeration, it was difficult to find well-debated languages, so I resorted to fieldwork data and to extracting the necessary information from descriptive grammars. For this reason, my data contains several pairs (or even larger groups) of genetically closely related languages. Quite often, when I was trying to explain something in one language, I ended up comparing it to its relatives, and it brought new insights. Sometimes I came across a language and was unable to find literature on its causative constructions, but I realized that its relative is much better studied. Instead of eliminating languages, I kept them in the sample, as they still illustrate my line of reasoning, and no representativeness constraints prevent me from keeping all the languages I find fitting. The whole list of languages used throughout this work can be found in the Index of Languages.

This thesis stems from the question "What would a language do when forced to add

[9]The first public demonstration of the method I use in this thesis – or of its preliminary idea – was the talk at the 17th Conference on Typology and Grammar for Young Scholars (Generalova 2020h). I am especially grateful to the audience for constructive criticism and valuable comments on individual languages, and to Peter Arkadiev for pointing out the inconveniences in the works of Ranko Matasović and for suggesting additional literature to me. This talk is referenced here mainly for bookkeeping purposes because the method presented in the thesis has significantly evolved since then.

an argument to the transitive clause?" A logical but not very informative answer would be: "Use one of its means to encode it". Thus, the goal of this chapter is to discover, describe, and discuss how exactly it can be done.

Working within the functionalist perspective, I would like first to assess the means of expressing arguments in three-argument causative constructions within a language. I would like to see what other syntactic positions and/or semantic roles receive the same marking as the elements of causative constructions. Relating causative constructions to other types of sentences in a language would help clarify how exactly the arguments of causatives are perceived and construed. I assume that there will be a closed set of those strategies, which is the enumeration I am looking for.

The easiest possible strategy for marking arguments in a three-argument causative construction would be to reuse a strategy that already exists in a language. The first candidate would be a non-derived three-argument construction, namely, a ditransitive. So, if a language has ditransitive constructions, it is predictable that the same patterns would be used for three-argument causative constructions. In this case, two ways of argument marking are theoretically possible, see (27) and (28).

(27) $E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{ditr}\ R_j^{ditr}\ T_k^{ditr}$

(28) $E_i^{caus}\ C_k^{caus}\ A_j^{caus}\ :\ A_i^{ditr}\ R_j^{ditr}\ T_k^{ditr}$

By definition of the causative construction (see Haspelmath and Müller-Bardey (2004) and Dixon (2000) cited in Chapter 1), the effector (the causer) receives the same marking as the actor of monotransitive and ditransitive constructions. However, no such restriction is, by definition, applied to other participants; hence the two options. The strategy illustrated in (24) is described and discussed in Section 6.1.1.1.1, while the strategy (28) receives special attention in Section 6.1.2.3.

The second type of strategy found in three-argument causative constructions would have two markers used in prototypical transitive constructions and a third marker that appears elsewhere in a language. In contrast to the previous case, this type of strategy can be related to other constructions of a given language but not to a single one. In other words, all three markers can mark arguments and appear in a given language, but no other construction is found where all three appear together.

(29) $E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{tr}\ U_k^{tr}\ :\ P_j^{other}$

A template for such a construction is shown in (29). The markers that are used for the effector (causer) and the affectee of the causative construction are the same as in a transitive construction (see *i* and *k*). The one used for the causee *j* appears on some participant *P* in another construction. There is a variety of constructions that can serve as sources of these markers. One of the goals of this chapter is to trace similarities between these constructions in terms of why all of them have something in common with three-argument causatives.

(30) $E_i^{caus}\ C_k^{caus}\ A_j^{caus}\ :\ A_i^{tr}\ U_k^{tr}\ :\ P_j^{other}$

Theoretically, yet another pattern of a three-argument causative can be related to two other constructions of a single language, see (30). In this scenario, the marker *i* for the effector (causer) is, as usual, the same as the one for the actor in transitive constructions. However, the marker for the undergoer of a transitive construction *k* is used not for the theme of the causative (as in (29)), but for the causee. The theme of the causative construction is, in turn, marked like some participant in another construction.

In practice, however, the realization of this pattern is rather questionable. However, some patterns in world languages provide some data that prevents discarding this type of constructions.

Finally, the third option for argument marking in three-argument constructions would involve only two markers. Some two participants thus would receive (seemingly) identical coding. Again, theoretically, there are two possible options, see (31) and (32).

(31) $E_i^{caus}\ C_i^{caus}\ A_k^{caus}\ :\ A_i^{tr}\ U_k^{tr}$

(32) $E_i^{caus}\ C_k^{caus}\ A_k^{caus}\ :\ A_i^{tr}\ U_k^{tr}$

The option demonstrated in (31) assumes that the causer and the causee receive identical marking, while the theme is marked differently. Since, by definition, the causer is intended to receive the PSA-like marking in the clause, this strategy somehow contradicts the initial definition. In reality, causative constructions having argument patterns like in (31) are attested. They are shown and extensively discussed in Section 6.2.2.4.

The second option (32) is more widespread. It is attested in languages with the so-called differential object marking (Section 6.2.3) and multiple (double) object constructions (Section 6.2.1). These cases are presented in Section 6.2.3 and Section 6.2.1.

All the subsequent sections of this chapter follow the same basic plan. First, language data is presented, and insights from descriptive studies are given. Next, these ideas are analyzed within the RRG and conform to the assumptions made in the current thesis. This analysis leads to the formulation of the distinctive properties of each construction, expressed as features necessary to describe it correctly and distinguish it from other strategies. In traditional RRG analysis, these data are structured into constructional schemas (Section 2.7), which are, in fact, provided at the end of each section. The chapter concludes with a discussion of all constructional schemas to trace further generalizations and unify axes of variation. This step leads to a constraint-based (Chapter 8) definition of the variety of three-argument causative constructions, which in turn is an important step towards their implementation.

Chapter 6

Argument structure and linking in each strategy

6.1 Strategies without doubling

6.1.1 Causee like Recipient & Causee like Demoted Agent

The present chapter opens with the description of two constructions that have received the biggest attention in prior studies: **Causee like Recipient** and **Causee like Demoted Agent**. First, I briefly overview each strategy. Next, I present classic analyses of Kannada (Dravidian), a language that was mentioned in several early studies on causative constructions (Cole 1983; Foley and Van Valin 1984). This language has both constructions, **Causee like Recipient** and **Causee like Demoted Agent**, and their competition was the main focus of prior analyses (starting from Hyman and Zimmer (1976)). Afterward, I discuss Bashkir (Turkic), a language also having both causative constructions in question. What differentiates Bashkir from Kannada is that it marks prototypical instruments and demoted agents in passives differently, thereby shedding light on principles overlooked in prior scholarship. Finally, I discuss the applicability of prior analyses to the new data and suggest some additions to the rules of linking in causative constructions of type **Causee like Recipient** and **Causee like Demoted Agent**.

6.1.1.1 Strategies presented

6.1.1.1.1 Causee like Recipient The strategy **Causee like Recipient** comprises three arguments marked identically to the participants of a ditransitive clause. The causee receives the same case as the recipient (usually dative, but not necessarily, e.g., allative in Estonian, cf. (33) and (34)).

(33)	Pille	näi-ta-s	Maarika-le	pilte	Estonian
	Pille.NOM	see-CAUS-PST.3SG	Maarika-ALL	picture.PAR.PL	

‘Pille showed Maarika the pictures.’ Vihman 2004, p. 108, (4.16b)

(34) Estonian

Tartu	ülikool	anna-b	üliõpilase-le	hea-d	võimalus-t
Tartu.GEN	university.NOM	give-3SG	student-ALL	good-PAR	opportunity-PAR

ennast	igakülgselt	arenda-da
oneself	many-sided	develop-INF

‘The University of Tartu offers a good opportunity to develop one’s interests in many ways.’ Pool 2010: 239 cited in Tamm and Kivik 2017, p. 270, (1)

As the most agentive participant and the highest argument in the logical structure, the causer becomes Actor (in macrorole terms) and PSA (in syntactic terms). The affectee, as the lowest-ranking participant, becomes the Undergoer and the object. The third argument, in RRG terms, receives no macrorole and becomes marked with the dative case since dative is the “default” (elsewhere) case for non-macrorole core arguments (Van Valin and LaPolla 1997, 533, 583). Van Valin and LaPolla (1997, 583) hypothesize that this pattern should be “the most frequent cross-linguistically”, following Comrie (1976) where this claim was first made.

Again, I would like to draw attention to possible terminological (or even conceptual) under-specification in this claim: should it be dative or the case used for non-macrorole arguments in underived ditransitive constructions? Some basic logic would support the second option. If a language has prototypical underived ditransitive constructions, it already has a way to encode the non-macrorole core argument. I believe that for reasons of economy, a language would demonstrate the same argument-marking patterns in syntactically similar constructions. Thus, featuring dative or any other case, the pattern attested in ditransitive is the first candidate to be reused in other three-argument cores, for instance, causatives with three core arguments, where the causee is a non-macrorole participant. This is exactly where the strategy **Causee like Recipient** originates from and why it is suspected to be extremely (if not the most) frequent.

However, if the causative construction is somehow different from the ditransitive one, the usage of a different pattern can be expected. Differences in syntax, semantics, and pragmatics of causative constructions, as well as features of individual participants that lead to variation in three-participant causative constructions, will be examined in detail throughout this chapter. My findings in this direction are summarized in the Conclusion (Chapter 7).

6.1.1.1.2 Causee like Demoted Agent

The strategy **Causee like Demoted Agent** receives attention in Van Valin and LaPolla (1997) primarily as an alternative to **Causee like Recipient**.

(35) a.

Je	fer-ai	mang-er	les	gâteaux	à	Jean	FRENCH
1SG.NOM	make-FUT.1SG	eat-INF	the.PL	cakes	DAT	John	

'I will make John eat the cakes.' Van Valin and LaPolla 1997, p. 582, (9.77b)

b.

Je	fer-ai	mang-er	les	gâteaux	par	Jean
1SG.NOM	make-FUT.1SG	eat-INF	the.PL	cakes	by	John

'I will have John eat the cakes.' Van Valin and LaPolla 1997, p. 533, (9.22b)

c. [**do**′(1SG, ∅)] CAUSE [**do**′(Jean, **eat**′ (Jean, gâteaux)) & BECOME **eaten**′(gâteaux)]

Van Valin and LaPolla 1997, p. 583, (9.78b)

The analysis states that in examples like (35b), the French *par*-phrase is an adjunct, judging by the fact that it can be omitted. Thus, (35b) is a construction with *a two-argument core.*

Interestingly, in RRG, both syntactic constructions (35a) and (35b) are described as having identical logical structures, as in (35c). Thus, RRG postulates that syntactic differences between these constructions originate not from the differences of the predicate but from the arrangement of arguments with the same predicate.

Of all possible ways of adjunct marking, French uses the *par*-phrase, which is also used for demoted agents in passive constructions (see Van Valin (2005, 535) notes; cf. *by*-phrase in English). Importantly, it is not the same preposition that is used for prototypical instruments. As shown in (25) and discussed in Section 5.3, *par* can encode abstract instruments (e.g., 'prove sth *with* examples'), but not the physical ones.

The The French causatives shown above are clearly biclausal and thus fall outside the scope of the present study. However, the mechanism of participant demotion attested there can be related to the one found in morphological causatives. The actor of the matrix logical structure, i.e., the causer, outranks the actor of the embedded clause, i.e., the causee, having more actor-like properties. It thus becomes the actor of the whole causative sentence. The second actor is then demoted to the periphery. It seems only logical that it receives the same marking as another kind of demoted agents, i.e., in passive constructions.

6.1.1.1.3 Data: Kannada

One of the languages having **morphological** causatives of both types **Causee like Recipient** and **Causee like Demoted Agent** is Kannada. Kannada is a Dravidian language with rich case morphology. According to Sridhar (1976, p. 134, (16b)), it encodes recipients with the dative case. Demoted agents of passive constructions, as shown in Dryer (1982, p. 313, (24)-(25)), are marked with the instrumental.

Thus, morphological causatives in Kannada demonstrate clear cases of **Causee like Recipient** (36a) and **Causee like Demoted Agent** (36b) constructions.

(36) a. Avanu-∅ nana-ge tīy-annu kud-is-id-anu — KANNADA
3SG-NOM 1SG-DAT tea-ACC drink-CAUS-PST-3SG.M
'He made me drink tea.'

b. Avanu-∅ nann-inda tīy-annu kud-is-id-anu
3SG-NOM 1SG-INS tea-ACC drink-CAUS-PST-3SG.M
'He had/let me drink tea.'

Cole 1983:121 cited in Foley and Van Valin 1984, p. 384, (vi)

Similarly to French, the main focus of prior analyses of these constructions in Kannada was not on describing each of them individually but rather on explaining why both coexist in a single language and what distinguishes them. Again, it seems that this line of analysis is due to the strong claim made in Comrie (1976) that participants marked with cases other than the case used for indirect objects can appear in a causative construction only when all positions to the right are filled. In other words, Comrie predicted that constructions like (36b) should not exist and launched research in many languages aimed at disproving this claim.

Foley and Van Valin (1984, 384–385) cites Kannada as an example of a language with two options for causee coding. Examples (36) constitute a minimal pair to show that both strategies are available for the same verb with the same NPs. This proves that the difference between the two ways of marking is actually the difference between two syntactic constructions and is not due to any lexical properties[10].

The main line of Kannada analysis given in Foley and Van Valin (1984) is the same as the one suggested for French. From the syntax-to-semantics perspective, two types of linking are suggested. Crucially, both constructions link to the same logical structure.

Importantly, Kannada is one of the languages that use the same marker for prototypical instruments and demoted agents in passive clauses. As already highlighted in Section 5.3, not all languages exhibit the same syncretism.

6.1.1.2 Analysis 1: The volitionality hypothesis

The main claim in Foley and Van Valin (1984, 384–385) is that the causees in (36) differ in their volitionality[11]. According to that, the instrumental causee is "the volitional controller of the action". The dative causee, in contrast, lacks volitionality.

This claim reflects observations in Cole (1983, 121). Commenting on the same exam-

[10]In fact, this minimal pair is not totally fair, although absolutely grammatical. There exist some lexical properties of the verb and of the nouns that influence the choice between the two strategies.

[11]I use the term volitionality like it is used in Van Valin and Wilkins (1996), which, in turn, takes over Fillmore's Case Grammar.

ples, the author notes that sentences like (36a) "are appropriate when the complement subject is, for instance, an animal or a child, that is, an individual to whom volition is not imputed". In contrast, the strategy with instrumental marking, like in (36b), is appropriate when the causee "is a mature adult". Animals and children are often opposed to "mature adults" on the scale of animacy, which, in turn, is closely related to volitionality.

However, this analysis requires some precision. In fact, the volitionality of the causee might already be a consequence of a more general principle. For easier comprehension, Foley and Van Valin (1984, 384–385) provides different English translations for (36a) and (36b) (note that in Cole (1983, 121) both sentences (36) have the same translation 'He caused me to drink tea'). The Kannada sentence with the dative causee (36a) is translated with the English verb 'make' corresponding to a construction where the causee is forced to perform the action.

Also, English 'make'-causatives are implicative, i.e., the action of the causee is implied and must necessarily take place. Implicative causative constructions cannot be followed by a counterfactual statement like **He made me drink tea, but I refused.* In Copley and Wolff (2014), this feature is called *defeasibility* (see comments in Section 3.10.2.1). In a recent paper on the semantics of English causatives, Nadathur and Lauer (2020, 9) formulate this as a sufficiency hypothesis: "given the occurrence of the cause, the occurrence of the effect was guaranteed/inevitable, in a weak sense to be made precise". Note that being implicative (or non-defeasible) is a property of the causative predication (or construction) as a whole, while volitionality is a property of an argument. These two parameters are strongly related but not identical; see discussion in Section 6.1.1.4.

The Kannada sentence with the instrumental causee (36b) is suggested to be translated with either 'have' or 'let.' English 'have'-causatives are implicative (like 'make'-causatives), but the causee is claimed to be volitional (in contrast to 'make'-causatives). Cf. the comment on a 'have'-causative sentence in Nadathur and Lauer (2020, 3): "[The causer] issued a straightforward order or instruction to [the causee], which they then obeyed without resistance or objection". This latter part, concerning the lack of resistance or objection, is linguistically conceptualized as the non-volitionality of the causee. The fact that English 'have'-causatives are implicative has also been confirmed by Van Valin, personal communication.

English 'let'-causatives are different from both 'make'- and 'have'-causatives because they are **not** implicative. Once followed by a counterfactual statement, they do not lead to unacceptable sentences: *The mother let the child drink tea, but he refused.* 'Let' has a permissive meaning, which seems to entail the volitionality of the causee. However, according to Nadathur and Lauer (2020, 27) *inter alia*, 'let' "can also occur in a range of contexts involving non-volitional effects". Van Valin (personal communication) shares this view of the English 'let'-causatives. The idiom *let go* as in *The policeman let go of the prisoner* suggests that the causee does not have much volitionality. Hence, the type of causation is more of an order rather than a permission. More prototypical constructions, such as *The teacher let the student quit the room, but he refused*, show that the volitionality of the causee has little relevance for the causer. In other words, the permission issued by the causer might have some or no effect on the causee. Whether the causee performs the caused action or not is not within the causer's control. Thus, English 'let' is not necessarily combined with either type of causees (volitional or non-volitional), whereas 'make' and 'have' require a single kind each.

Going back to Kannada examples, I can now add some precision to the analysis: sentences with dative causees (36a) are implicative with non-volitional causees; sentences with instrumental causees (36b) are either implicative with volitional causees, or non-implicative. This second dimension has not been included in this analysis before, but will prove important when comparing Kannada with other languages.

6.1.1.3 Analysis 2: The Agentivity Implicature hypothesis

The volitionality hypothesis explains the choice of a given case in each Kannada causative construction: volitional causees receive INS, and non-volitional causees receive DAT. A broader question is why this competition occurs at all. Since the competition of these two strategies is fairly regular across languages, its explanation in terms of lexical idiosyncrasies would be insufficient. There should be a general rule that triggers different argument marking.

This question receives due attention in RRG studies. In Van Valin and Wilkins (1996) and Van Valin (2018a), the authors postulate an "agentivity implicature". This term and concept were first introduced in Holisky (1987), a study of intransitive subjects in Tsova-Tush (also known as Batsbi, Nakh-Daghestanian). Put simply, the implicature situation is formulated as follows: "with some verbs, agentivity is not a lexical property but rather an implicature involving human (and high animate) effectors… In the implicature case, human effectors may be interpreted as acting willfully and volitionally in the absence of evidence to the contrary; the agentive interpretation can be blocked[12]" (Van Valin 2018a, 125). It is grounded on a pragmatic principle: "You may interpret effectors and effector-themes which are humans as agents (in the absence of any information of the contrary)" (Van Valin and Wilkins 1996, 309). One can clearly see that a key point in this reasoning is the volitionality of the participant. However, the implicature approach provides more precise pragmatic features that condition participants to be understood as volitional or not.

The two possible interpretations of a sentence are thus formulated as *blocking* vs. *fa-*

[12]Again, this reasoning is not completely free from some lexical assumptions. The implicature arises only "with some verbs", apparently, having some common lexical properties, and tackles the animacy of the participant (which is, in turn, a lexical property of nouns). Thus, when drawing this general conclusion, one must remember that it is a trend rather than a rule applicable to some language items to a different extent.

voring of the agentivity implicature. *A block* (also called *a disfavoring*) of the implicature in a **causative** construction means that interpreting the causee as an agent is impossible (or not normal) (Van Valin and Wilkins 1996, 311, Van Valin 2018a, 127). This is directly related to low volitionality. In contrast, *favoring* the agentivity implicature means that the causee can be interpreted as an agent. Usually, in this case, it is highly volitional.

In Tab. 11, Van Valin (2018a, 127) summarizes ways of morphological marking of the causee in relation to how the agentivity implicature is resolved.

Language	Disfavoring AGENT Implic.	Favoring AGENT Implic.	Base verb: Intransitive or Transitive
French	Dative	**passive agent (*par*)**	Transitive
Japanese	**Accusative**	Dative	Intransitive
Basque	**Absolutive**	Dative	Intransitive
Kannada	Dative	**Instrumental**	Transitive
Bolivian Quechua	Accusative	**Instrumental**	Transitive

Table 11: Alternations in causee coding in causative constructions (reproduced with some adjustment from Van Valin 2018a, 127)

However, the situation in real languages is usually more complex. One important thing initially not represented in Tab. 11 is the neutralization of the agentivity implicature.

As noted by Holisky (1987), in many cases, the contrast between two coding options is realized as a privative opposition.

> For a large number of verbs . . . the ergative is unmarked and can convey agentivity or non-agentivity. The form with nominative, as the marked member of the opposition, is always interpreted non-agentively. For a smaller number of verbs . . . however, the marking is reversed: the ergative is the marked member of the opposition and is used to convey agentivity on the part of the subject. The nominative is neutral, expressing nothing in particular about agentivity. Holisky (1987, 116)

Developing this idea, RRG postulates the same type of opposition for French periphrastic causatives. Thus, dative causees (introduced by the preposition *à*) are unmarked, i.e., lack markedness. The other French causative strategy, using the preposition *par*, is the marked member of the privative opposition. Hence, it requires a specific way of resolving the implicature, namely, it must be interpreted as favoring it.

The situation in Japanese is somehow opposite to French (according to Van Valin (2018a, 126)): "the case neutral to the agentivity implicature is the dative, while the accusative blocks it". So, the marked members of the privative opposition in Japanese and

French are associated with different strategies of the implicature resolution. I added this information to Tab. 11, emphasizing in boldface the **marked** members of the privative opposition.

From this, one more observation can be drawn: the dative case tends to encode the neutralization of the implicature, be it the opposition of the marked favoring or the marked blocking (consider non-boldface cases in Tab. 11). As noted in Van Valin and LaPolla (1997, 589), this goes in line with the RRG analysis of the dative case as “the default case for (non-macrorole) core arguments”. As the term ‘default’ can be used in a very wide range of contexts and oppositions, I prefer the term ‘elsewhere case’ to capture its tendency to be used in contexts without specific marking.

The coexistence of two three-argument causative strategies with different causee coding should then be analyzed using the agentivity implicature hypothesis as a multi-step algorithm. First, one needs to ensure that the construction is eligible for the agentivity implicature. In the logical structure of the causative verb (with any possible language-specific and lexeme-specific idiosyncrasies), there must be two participants that could potentially be construed as actors. Second, one needs to ensure that the language is sensitive to the resolution of this implicature. Usually, the presence of two implicative causative constructions would be the reason to suspect this sensitivity. Third, one needs to determine which member of the privative opposition is marked. A language might need to explicitly encode either the favoring of the implicature (like French) or the blocking of it (like Japanese). Finally, it comes to coding options.

The marked member of the opposition receives specific marking based on several factors: its macrorole status, general case assignment rules, language-specific conventions, and any lexeme-specific idiosyncrasies. This marking process usually results in the encoding of the causee as the prototypical undergoer in the situation of implicature block. Conversely, when the construction favors the implicature, the causee tends to be coded in the same way as the prototypical instrument or the demoted agent found in passive constructions. In this situation, the unmarked member of the privative opposition generally receives ’elsewhere coding,’ as there are no particular rules governing its marking. Most frequently, this unmarked participant is assigned the dative case.

A preliminary scheme of an updated coding algorithm is shown in Fig. 8 below. In accordance with what has been described earlier, implicative causative constructions yield four different coding strategies (see Strategies 1-4 in Fig. 8). Two of them (those with unmarked members of the privative opposition, i.e., neutral towards the agentivity implicature) often end up being marked in the same way, usually with the dative as the ‘elsewhere’ case (see Strategy 2 and Strategy 3).

With respect to Strategy 4, the one explicitly favoring the agentivity implicature and thus involving a volitional causee acting in an implicative context, Tab. 11 appears to provide insufficient data. First, it does not list any language with several causative con-

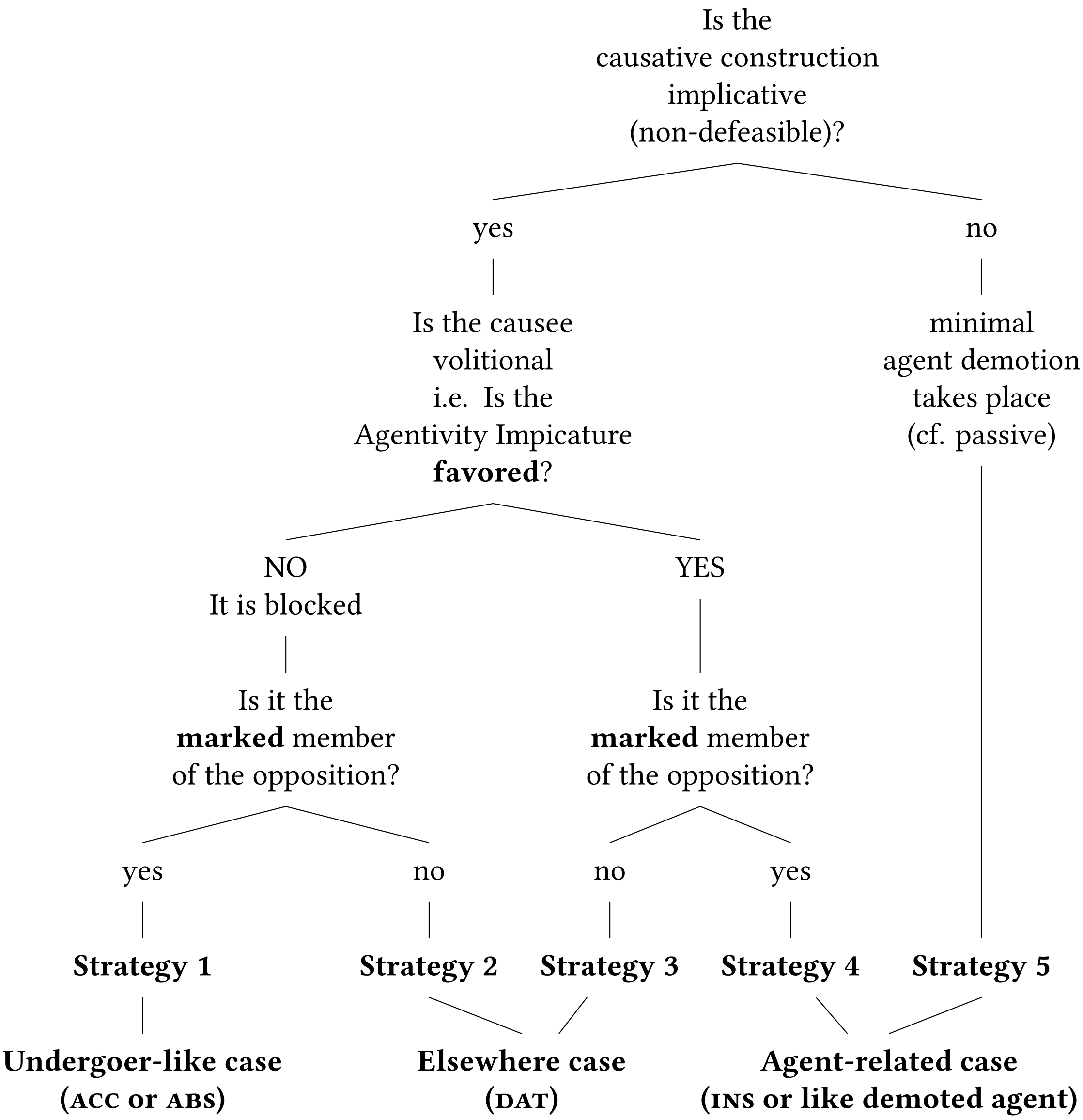


Fig. 8. An updated algorithm of causee coding depending on implicativeness and volitionality

structions based on intransitive verbs that would have a **marked** implicature-favoring strategy. (In both lines about intransitive bases, the 'favoring' cell has dative, which is assigned as the 'elsewhere' case.) One would predict instrumental appearing there, following the usual RRG rule for the instrumental assignment (because only effectors competing for the highest macrorole are involved in the agentivity implicature, as well as in the instrumental assignment rule) – see (37).

(37) Assign instrumental case to non-MR *b* argument if, given two arguments, *a* and *b*, in a logical structure, with (1) both as possible candidates for a particular macrorole and (2) *a* is equal or higher (to the left of *b*) on the AUH, *b* is not selected as that macrorole. Van Valin (2005, p. 110, (4.29))

Second, and closer to the focus of the present chapter, Tab. 11 does not make enough distinction between coding options used for Strategy 4 with transitive bases. The French data clearly indicate that the causee is marked like the demoted agent in a passive construction and not like the instrument. As for Kannada and Bolivian Quechua, Tab. 11 does not specify the scope of the instrumental case. In Kannada, the instrumental case is known to mark demoted agents in passive constructions (Dryer 1982, p. 313, (24)-(25)). Hence the question: is Strategy 4 in Kannada using the **Causee Like Demoted Agent** or **Causee Like Instrument** pattern? Comparing Kannada to French, one would hypothesize that Strategy 4 yields **Causee Like Demoted Agent** pattern, nonetheless, more data would have been helpful.

Finally, it is unknown how Strategy 4 relates to what in Fig. 8 is called Strategy 5. The two parameters – implicativeness and volitionality – have not been combined in a single decision tree before. Therefore, the branch concerning non-implicative causatives is barely developed, resulting in a single Strategy 5. From (36b), one knows that causative constructions that might have a non-implicative reading end up having an instrumental causee in Kannada. Hence, Strategy 5 ends up in the same block as Strategy 4 in Fig. 8.

Given this extended protocol, one can then approach Kannada causatives (36) and offer a refined explanation of their differences. If both (36a) and (36b) are implicative, we learn that Kannada is a type of language that cares about marking the favoring of the agentivity implicature. Therefore, the instrumental case, which is expected for implicature favoring, competes with the dative, assigned as 'elsewhere case'. The question is: what if one of these constructions is not implicative? One might predict that there is minimal agent demotion in this case since the agent in a non-implicative causative construction is neutral to the parameter of volitionality. Hence, one would predict that the causee receives a marking similar to that of other demoted agents, e.g., those in passive constructions.

In a cross-linguistic perspective, it is necessary to explore the contrasts between implicative and non-implicative causees further to determine how exactly argument-marking strategies in non-implicative causatives are realized and whether they are different from patterns attested in implicative causatives, favoring the agentivity implicature. These questions are addressed in the next section with the help of evidence from Bashkir and other language varieties.

6.1.1.4 Bashkir

Bashkir[13] is a Turkic language spoken in the Bashkortostan Republic (Russian Federation). Unlike Kannada, it codes prototypical instruments and demoted agents in passive constructions differently, see (39) vs. (40) below. Thus, Bashkir sheds light on the peculiarities of the strategy **Causee like Demoted Agent** without confusing it with any other one.

6.1.1.4.1 New data In Bashkir, verbal causative derivations are very productive. Although several causative suffixes can be attached to intransitive or transitive bases, according to Perekhval'skaya (2017, 238), their combination with the verbal stem is normally motivated by morpho-phonological reasons. The choice of argument-marking pattern does not depend on the suffix used.

There are two causative constructions occurring with transitive verbal bases, see (38). This is a minimal pair of examples featuring the same lexemes used in two different argument-marking strategies[14].

(38)	a.	Babaj	ul-ə-nan	xat-tə	uqət-tər-a	BASHKIR
		old.man	son-POSS.3-ABL	letter-ACC	read[16]-CAUS-IPFV△	
		'The grandfather asks his son to read the letter.'□				
	b.	Babaj	ul-ə-na	xat-tə	uqət-tər-a	
		old.man	son-POSS.3-DAT	letter-ACC	read-CAUS-IPFV	
		'The grandfather lets his son read the letter.'□				

Perekhval'skaya 2017, p. 244, (16a-16b)

Dative, unsurprisingly, is primarily used for recipients in ditransitive constructions. Therefore (38b) is a clear instance of the **Causee like Recipient** strategy type. The

[13]Bashkir is the language with which I engaged as a field linguist (Generalova 2016; Generalova 2014). Therefore, I began learning RRG and applying its method, which led to a series of talks (Generalova 2020b; Generalova 2019b; Generalova 2019c) and an article (Generalova 2020g). Even though these publications reflect the early stages of my research, the line of analysis presented in the present thesis has evolved much further since the last publication and is barely comparable to any of these prior works. The solution presented in the thesis is more up-to-date than any prior version.

[14]In fact, most verbs usually demonstrate strong preferences towards one of these strategies. See Perekhval'skaya (2017) for details.

[16]According to Perekhval'skaya 2017, the chunk *uqət* consists of the verbal root *uqə* and a fossilized causative marker *t*. However, the verb *uqət-tər-a* does not behave as a double causative and has only three participants. I modified the original glossing to avoid possible confusion.

other strategy, as shown in (38a), requires a causee marked with an ablative. One of the functions of this case is to mark demoted agents in passive constructions (Dmitriev 2008 [1948], 224, Yuldashev 1981, 145), see (39).

(39)

hin	jaqšə-lar-ðəŋ	tel-där-e-nän	maqta-l-də-ŋ	Bashkir
2sg	good-pl-gen	tongue-pl-poss.3-abl	praise-pass-pst-2sg	

'You were praised by the tongues of the best.'□ Dmitriev 2008 [1948], p. 224

Importantly, abl in Bashkir is not used for marking instruments. They are normally expressed by a phrase with the postposition *menän* 'with' (Yuldashev 1981, 135), see (40). Thus, the argument marking strategy shown in (38a) can be classified as type **Causee like Demoted Agent** and is definitely not **Causee like Instrument**.

(40)

ul	keše	beð-ðe	qotqar-ðə	jəlan-ɣa	tajaɣ-ə	menän	huq-tə	Bashkir
that	person	we-acc	save-pst	snake-dat	stick-poss.3	with	hit-pst	

'That person saved us, [s/he] hit the snake with a stick.'□

Ovsjannikova et al. 2019

Perekhval'skaya (2017) offers an extensive comment on the contrast between causative constructions demonstrated in (38a)-(38b):

> < In (38a), > the old man discovers the contents of the letter by means of the son (e.g., he comes up with this solution because he has forgotten his glasses). The letter is most probably read aloud in the presence of the old man. In (38b), the son is interested in discovering the contents of the letter. He may read it on his own, in a place or time different from the situation of causation. Perekhval'skaya (2017, 244–245)

This observation highlights the opposition between an implicative (38a) and a non-implicative (38b) causative construction. It stands out that the action of reading in the **Causee like Recipient** (38b) construction is more independent from the activity of causation than in the **Causee like Demoted Agent** (38a) construction.

To underline the implicativeness of (38a), Perekhval'skaya (2017, 244) draws a parallel with an identical construction (41) in Mishar Tatar (a language variety genetically very close to Bashkir).

(41)

Renat	zexrä-dän	išek-ne	jap-tɤr-a	Mishar Tatar
Renat	Zukhra-abl	door-acc	close-caus-st.ipfv	

'Renat makes/ Zukhra close the door.'□ Bonch-Osmolovskaja 2007, p. 147, (4b)

Bonch-Osmolovskaja (2007, p. 147, footnote) states that in examples like (41), the event denoted by the base verb is implied (= must happen) and that the causing action cannot be contradicted. She explicitly states that it is impossible to continue the sentence (41) with something like "but Zukhra does not obey him". She also offers a Russian translation of this example, emphasizing that *Zukhra* is treated almost as an inanimate object (or force): "A more precise translation of (41) would be *'Renat is closing the door **by means** of Zukhra'* or *'Renat **sanctions** Zukhra to close the door'* " (boldface mine – VG).

This comment suggests that the construction in (41) is at the extreme of the permission scale: the causer is the primary instigator of the event, and its will influences the causee's actions. However, the causer does not *permit* (or *let*) the causee to perform the action with some degree of independence. Instead, the causer *orders*, i.e., limits the independence of the causee. This axis cuts across the volitionality of the causee. In other words, in causative constructions where the dimension of permissiveness plays a role, the volitionality of the causee does not matter.

Going back to Bashkir examples (38), one can thus state an opposition between an implicative non-permissive (38a) (parallel to (41)) and a non-implicative permissive (38b) construction. It is thus a different opposition than those demonstrated in Tab. 11.

According to Fig. 8, the implicative Bashkir construction should have a causee marked with a dative because the construction is implicative, and the agentivity implicature is neutralized. However, Bashkir implicative causatives receive ablative causee marking. The **Causee like Demoted Agent** strategy is predicted by Fig. 8 only in situations with an explicit favoring of the agentivity implicature, i.e., with a marked volitional causee. However, the study of Bashkir examples shows that this assumption does not hold.

6.1.1.4.2 An alternative analysis Since the agentivity implicature does not explain the distribution of cases in Bashkir causative constructions, we need to provide an explanation based on the specific rules for using each case. The markedness of constructions with the dative causee suggests that there must be a criterion for selecting the marked pattern over the unmarked one. In this section, I summarize claims from descriptive studies on Bashkir. I also use data from Kalmyk (a Mongolic[17] language spoken in the Republic of Kalmykia, Russian Federation) and insights from descriptive studies on the argument marking in Kalmyk three-argument causative constructions.

In descriptive studies, scholars discuss semantic categories that are important for distinguishing the two causative constructions (illustrated above with (38a) and (38b)) from each other. Some data demonstrate that it is not the volitionality of the causee that matters pragmatically, but its *affectedness*. In other words, it is not that important

[17] Most of the Russian scholars, including all the authors cited in my work, define a typological entity of a higher level, including Turkic, Mongolic, and Tungusic language families. This entity is called the *Altaic macro-family*, or, most often, Turkic, Mongolic, and Tungusic are called *Altaic languages*. The problem statement and the debate concerning the salience of this classification entity can be found in Starostin (1991). Most volumes by Western linguists, however, reject this notion, e.g., Pereltsvaig (2012). The volume Robbeets (2015) uses the term "Transeurasian" in its title and offers arguments in favor of the Altaic hypothesis. The latest volume on the problems of Turkic and Mongolic languages seems to be Rassadin (2019). Consider also Dybo (2017) for the summary of the current state of the debate.

to what degree the causee demonstrates properties of an agent but to what extent it is perceived as some kind of an undergoer.

In Bashkir, the causee never outranks the undergoer in having undergoer-like properties. Hence, when a causative verb is passivized, it is the accusative participant, not the dative one, that becomes the PSA. No other option is possible, whatever the causee marking. Consider (42) with the verb 'cause-to-see', which demonstrates a strong tendency to be the head of causative constructions with a dative causee.

(42) BASHKIR

täw-ðä	unəŋ	urən-ə	Bäläkäj	Boɣaðaq	kül-e	buj-ə-nda
first-LOC	that.GEN	place-POSS.3	small	Bugodak	lake-POSS.3	length-POSS.3-LOC

kür-hät-el-gän	inde
see-CAUS-PASS-PTCP.PST	yet

'Then its place was shown near the lake Small Bugodak.'□ Ovsjannikova et al. 2019

In (42), the verb 'see' is both causativized and passivized. Only one participant is overtly expressed in this sentence, *urən-ə* 'place'. Clearly, it is extremely improbable for this noun to be the Causee and not the Theme.

Thus, the Causee is never **the** Undergoer in Bashkir causative constructions. Nevertheless, it can be close to the undergoer to varying degrees. According to descriptive studies, in both examples (43)-(44), the causer is interested in making the causee involved in the caused action. In terms of *affectedness*, the causee in (43)-(44) is more affected by the action of the causer than in (45), hence DAT vs. ABL.

(43) BASHKIR

Bala	besäj-gä	höt-tö	es-er-ä
child	cat-DAT	milk-ACC	drink-CAUS-IPFV

'The child feeds milk to the cat.'□ Perekhval'skaya 2017, p. 241, (10b)

(44) BASHKIR

Dilä	beð-gä	üð-e-neŋ	kejäw	jeget-e-neŋ	foto-hə-n
Dila	we-DAT	3.SG-POSS.3-GEN	mate	boy-POSS.3-GEN	photo-POSS.3-ACC

kür-hät-te
see-CAUS-PST

'Dila showed us the photo of her boyfriend.'□ Perekhval'skaya 2017, p. 243, (14)

(45) BASHKIR

Ataj	bala-lar-ðan	jeläk	jəj-ðər-a
father	child-PL-ABL	berry	pick.up-CAUS-IPFV

'The children are picking up berries upon request (order) of the father.'□

Perekhval'skaya 2017, p. 245, (17b)

An extensive discussion of the affectedness of the causee can be found in Say (2009) (although the paper explores the Kalmyk language, the examples are constructed identically to the Bashkir ones, cf. (46a) // (43) and (46b) // (44)).

(46) a.

ekə	ürə-n-d-än	xašə	id-ül-ǯä-nä	KALMYK
mother	child-EXT-DAT-POSS.REFL	porridge	eat-CAUS-PROG-PRS	

'The mother feeds the porridge to the child.'□ Say 2009, p. 406, (25)

b.

Badma	gerg-n-ännj	zurəg	nan-də	üz-ülə-v	KALMYK
Badma	wife-EXT-GEN.POSS.3	image	me-DAT	see-CAUS-PST	

'Badma showed me a photo of his wife.'□ Say 2009, p. 406, (28)

Say (2009) makes the following observations. (One can note yet another usage of the preposition "by means of" that appears consistently in comments by different authors.)

> What I am talking about is that the causer's goal in these cases is not the change of state of the participant coded as the direct object, but a certain impact on the causee. It is especially evident for verbs of the mental sphere – it is clear, for example, that in (46b), the photo does not undergo any change during the causing event; there is only the change of state of the participant to whom it is shown. In (46a), the porridge, of course, undergoes a significant change, but the idea of the causation is not to make the porridge vanish by means of the child but to feed the child, i. e, make him not hungry.
> Say (2009, 407)

The analysis using the *affectedness* feature explains why the causee sometimes is closer to the Undergoer than expected. However, the same participant is selected as the Undergoer in both causative constructions. Thus, Bashkir is different from languages with "variable undergoer selection" (Van Valin 2005, 111) in RRG terms. No principle concerning the second-lowest argument in syntactically accusative languages has been attested so far. Therefore, the explanation from the Undergoer perspective for languages such as Bashkir, Kalmyk, and Mishar Tatar is quite unexpected.

What is also found in descriptive studies is the analysis of the causee as a possible instance of a non-prototypical recipient. Indeed, the strategy with the dative causee is preferred with verbs having a recipient/beneficiary role, as shown (43)-(44) for Bashkir and (46) for Kalmyk. Perekhval'skaya (2017) notes that verbs of ingestion (like in (43)) and verbs of experiencing (like in (44)) are the most common predicates to occur in constructions of type **Causee like Recipient** (i.e., with the dative causee marking). Moreover, data from Ovsjannikova et al. (2019) suggest that the condition works in the other direction as well: verbs of ingestion and experiencing occur in constructions of type **Causee like Recipient** more often than in constructions of type **Causee like Demoted Agent**. The same observations are made by Say (2009) about causative constructions in Kalmyk. This could be a hint that similar mechanisms underlie the formation and use of causative constructions across different groups of Altaic languages.

Commenting on Bashkir, Perekhval'skaya (2017, 241) notes that the strategy with dative causee marking is mostly used when the action is “a benefit” for the causee (she also notes that the benefit can be negative, i. e, a hindrance). The notion of benefit combines both semantic properties named above: similarity to a recipient and affectedness. A beneficiary is normally a recipient in an action of some non-physical transfer, being affected by changing its state from lacking something into possessing something (or the other way round in case of a “negative benefit”).

It is important to note that the choice of the verb does not condition the explanation of the variable marking. Although there are verbs that tend to be used predominantly in constructions with dative causee marking, some other verbs (e.g., ‘read’ in (38)) can occur in either construction. The existence of examples like (38) requires, thus, a pragmatic explanation of the argument structure variation. Thus, the feature of *benefit* is assigned to the causee only in some situations on pragmatic grounds, yielding the construction with the dative as the marked member of the opposition.

The remaining question is the assignment of the *benefit* feature in constructions with ablative causee marking. It looks from the comments previously offered to examine this strategy that it is the causer who benefits from the whole causative event. Consider the comment in Perekhval'skaya (2017, 245) repeated: “[In (38a),] the old man discovers the contents of the letter by means of the son (e.g., he comes up with this solution because he has forgotten his glasses)”. It stands out clearly that the goal of the causative event is to make the content of the letter known to the old man. Whether the son is affected by this action is irrelevant. Given that, it is appropriate to formulate the contrast between the Bashkir strategies **Causee like Recipient** and **Causee like Demoted Agent** in terms of benefiting. The default assumption for Bashkir (and, seemingly, Mishar Tatar and Kalmyk) is that the causer benefits from having the caused action done. Therefore, the causee is most often marked as the demoted agent. The key property of the causee acting as a force for performing the action is, in fact, its (potential) agentivity. To ensure that the caused action takes place (which is obligatory, as Mishar Tatar data show), the causer must ensure that the causee is able to perform it. In this light, the usage of the same marker for the causee and the demoted agent appears very consistent: both of them are highly agentive, but are not agents in the situation described by passive or causative sentences.

Thus, one can formulate the distribution of the feature ‘benefit’, see Tab. 12. The strategy **Causee like Recipient** is the marked member of this opposition, therefore, it is emphasized with boldface in Tab. 12.

This opposition reflects observations from the descriptive studies but does not help make RRG-inspired predictions about these strategies. Some of the examples can be

Strategy type	Causer	Causee
Causee like Recipient (DAT)	-ben	+ben
Causee like Demoted Agent (ABL)	+ben	-ben

Table 12: The feature 'benefit' in Bashkir causative constructions

explained by postulating a special strategy for verbs that can take a recipient or beneficiary. These are verbs of ingestion and of physical and non-physical transfer. The special behavior of these verbs is not unexpected. Verbs of transfer are closest to prototypical ditransitives and thus are prone to exhibiting the same patterns. Verbs of consumption are also often somewhat different. Namely, the present work examines Lubukusu (Bantu, see Section 6.2.1.2.5), where verbs of consumption also trigger a marked pattern of a three-argument causative construction, which is similar to ditransitives.

However, not all Bashkir examples can be explained using this lexical approach. It is clear that there can be a pragmatic motivation for the marked choice of the beneficiary. Therefore, I suggest putting this information into the constructional scheme of Bashkir causatives, see Fig. 9.

6.1.1.5 Summary

In Section 6.1.1.5, I have presented two causative strategies: *Causee like Recipient* and *Causee like Demoted Agent*. I have previously offered two analyses of the competition between these strategies in a single language. I have shown that the axis of participants' volitionality cuts across the axis of the construction's implicativeness (non-defeasibility), once both play a role in an analysis of a given sentence. I have drawn attention to the syncretism of marking prototypical instruments and demoted agents in passive in some languages. To observe a clear case of the strategy *Causee like Demoted Agent*, I have studied the data from Bashkir, where this syncretism is not attested. I have also shown that the standard RRG analysis based on the Agentivity Implicature does not yield correct analyses of Bashkir sentences. Hence, I have determined that the *benefit* is the decisive semantic feature determining argument marking in Bashkir. Overall, the observations in this section show that the confusion between two functional positions – a prototypical instrument and a demoted agent – has obscured prior analyses. A thorough typological study aimed at delimiting syncretic situations from non-syncretic ones would be a reasonable next step in research.

Strategy:	Causee like Recipient
Relatable to:	passive
Argument encoding:	$E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{tr}\ U_k^{tr}\ :\ demA_j^{pass}$
Macrorole assignment:	$E_a^{caus}\ C_{nmr}^{caus}\ A_{ug}^{caus}$
Special features:	(1) unmarked, (2) Causer [+ben] Causee [-ben]

a. Causee like Demoted Agent

Strategy:	Causee like Recipient
Relatable to:	ditransitive
Argument encoding:	$E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{ditr}\ R_j^{ditr}\ T_k^{ditr}$
Macrorole assignment:	$E_a^{caus}\ C_{nmr}^{caus}\ A_{ug}^{caus}$
Special features:	(1) marked, (2) Causer [-ben] Causee [+ben]

b. Causee like Recipient

Fig. 9. Constructional Schemas for Bashkir causatives

6.1.2 Other ditransitive-like strategies

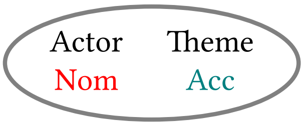
6.1.2.1 Description of the strategies

If there exists the strategy **Causee like Recipient**, there must be, perhaps, the strategy **Affectee like Recipient**. In fact, it also involves the set of markers used in non-derived three-argument constructions, i.e., ditransitives. Contrary to the **Causee like Recipient** strategy, the affectee would be marked as the recipient of a ditransitive construction, whereas the causee would be marked as the theme of the ditransitive construction, see (47) (same as (28); repeated below for convenience).

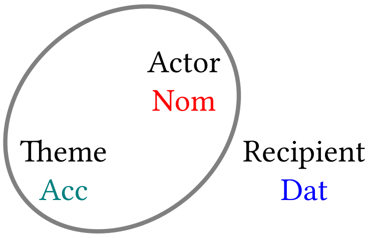
(47) $A_i^{ditr}\ R_j^{ditr}\ T_k^{ditr}\ :\ E_i^{caus}\ C_k^{caus}\ A_j^{caus}$

This syntactic marking implies a different macrorole structure as well. Namely, the causee would be the Undergoer, and the affectee would be the non-macrorole participant.

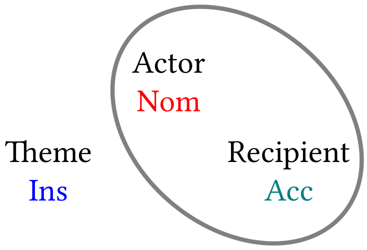
One should regard the causative strategies **Causee like Recipient** and **Affectee like Recipient** together with alignment patterns of basic ditransitive clauses. In fact, they *mirror* each other as indirective and secundative (Haspelmath 2005 after Guerrero and Van Valin 2004) alignments in ditransitive constructions when related to the monotransitive marking pattern. Consider Fig. 10 illustrating hypothetical ditransitive and causative alignments in a nominative-accusative language. In Fig. 10d entitled "Causative 1", cases used in the monotransitive construction (NOM and ACC) are assigned just like they are in Fig. 10b "Ditransitive indirective". Similarly, Fig. 10e "Causative 2" corresponds to the ditransitive secundative alignment demonstrated in Fig. 10c. Therefore, once related to monotransitive constructions, both Causative 1 and Causative 2 seem to be equally predictable, just like both ditransitive alignments – indirective and



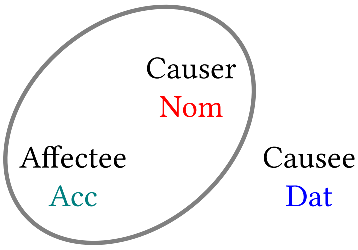
a. (Mono)Transitive alignment



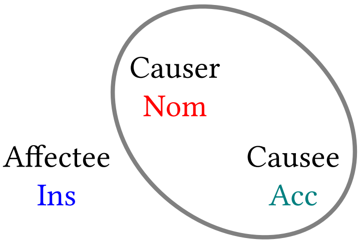
b. Ditransitive indirective



c. Ditransitive secundative



d. Causative 1



e. Causative 2

Language	Causative type	Ditransitive type	Causative strategy
Language A	Fig. 10d	Fig. 10b	**Causee like Recipient**
Language B	Fig. 10e	Fig. 10c	**Causee like Recipient**
Language C	Fig. 10e	Fig. 10b	**Causee like Theme**
Language D	Fig. 10d	Fig. 10c	**Affectee like Recipient**

f. Causative Strategies summarized

Fig. 10. Causative strategies with respect to causative and ditransitive alignments

secundative – are attested in nominative-accusative languages[18].

However, in the present work, I am interested in relating causative constructions not only to monotransitives but also to ditransitives. The terms **Causee like Recipient** and **Affectee like Recipient** result from this comparison and do not account for the underlying kind of monotransitive alignment. If a hypothetical language depicted in Fig. 10 had a single ditransitive construction with indirective alignment (Fig. 10b), Causative 1 (Fig. 10d) would be considered **Causee like Recipient** (both marked DAT), and Causative 2 (Fig. 10e) would be considered **Causee like Theme** (both marked ACC). Note that in this case, **Causee like Theme** $\neq$ **Affectee like Recipient**. In Fig. 10b and Fig. 10e, the Theme and the Causee have the same case, namely, ACC. However, the alignment does not state that the third member of each construction bears the same case. In Fig. 10b and Fig. 10e it is exemplified with DAT and INS respectively. If a language has a secundative ditransitive construction (Fig. 10c), causatives like Fig. 10e are considered **Causee like Recipient**. Causatives like Fig. 10d would be of type **Affectee like Recipient**. Once again, one cannot be sure that the remaining participants' cases are the same. All these options are summarized in Tab. 10f.

A closer comparison of ditransitive and causative alignments leads to the postulation of three ditransitive-relatable causative strategies: **Causee like Recipient**, **Causee like Theme**, **Affectee like Recipient** (see Tab. 10f). Grounding on these observations, one can formulate the relationship between ditransitive and causative assignments as follows:

(48) a. If both – the ditransitive and the causative – constructions follow *the same* alignment pattern (either indirective or secundative), the causee receives the same marking as the *recipient* of a ditransitive.

b. If both – the ditransitive and the causative – constructions follow *different* alignment patterns, two options are possible: (1) the causee receives the same marking as the *theme* of a ditransitive OR (2) the *affectee* receives the same marking as the *recipient* of a ditransitive.

This conclusion answers the question of why constructions of type **Causee like Recipient** are extremely more frequent than the other two types. First, they are attested in languages with any ditransitive alignment (while each of the strategies **Causee like Theme** and **Affectee like Recipient** is attested in languages with one type of ditran-

[18]The simplest support for this claim can be obtained in WALS (Dryer and Haspelmath 2013) by combining features "Alignment of Case Marking of Full Noun Phrases" (Comrie 2013) and "Ditransitive Constructions: The Verb *Give*" (Haspelmath 2013) on the same map. Of 137 languages taking part in the intersection, 30 languages have standard nominative-accusative monotransitive alignment and indirective ditransitive alignment, and 3 languages have nominative-accusative monotransitive alignment and secundative ditransitive alignment. The claim holds for languages with ergative-absolutive alignment as well: 13 of them have indirective ditransitive alignment, and 5 – secundative ditransitive alignment. Although this quick estimation does not take into account many other patterns of alignment distribution, it supports the possibility of another three-argument construction with any alignment in a language.

sitive alignment only). Second, and most importantly, it is not very plausible that a language would use different kinds of alignment for different three-argument constructions. What reasons might lead to this is a topic for upcoming research in the typology of alignment.

6.1.2.2 Data: Javanese

6.1.2.2.1 General concerns According to Dixon (2000), Javanese is suspected to have the **Affectee like Recipient** type of causative constructions. However, there is a strong doubt whether morphological causatives with transitive verb bases exist in Javanese at all.

A corpus study based on standard Javanese editorials and short stories attests only 5% (4 of 80 instances) of causatives-of-transitives, none of which is morphological (see Tab. 3 in Malihah (2014, 65)). These data show that transitive-based morphological causatives are very rare or even nonexistent in modern standard Javanese.

Besides, it is difficult to tell causative constructions apart from ditransitive and applicative constructions, since verbal markers in Javanese are extremely syncretic. This is the second major concern about Javanese causative constructions – maybe they belong to another type of constructions, where the pattern they exhibit is less uncommon.

However, some papers focusing on other varieties of Javanese report that causative-of/transitive constructions can still be present. Most sources (Suhandano 1994; Hemmings 2013; Nurhayani 2014) agree that morphological causatives in Javanese are built with the suffix *-ake*. This marker is extremely syncretic: it also builds applicatives in some other Austronesian languages and also appears on some other three-place predicates in Javanese and other languages. Note that the suffix *-ake* is often glossed as CAUS, which is only partially salient. In fact, it is a highly multifunctional morpheme. For this reason, I decided to follow the glossing tradition in Sofwan (2010) and give no interpretative gloss label to this morpheme, cf. the label 'AKE' in (49)-(50).

(49) Sri ng-uyak-ake asu-ne marang Bambang — JAVANESE
Sri ACT-chase-AKE dog-DEF to Bambang△
'Sri held out the dog to chase Bambang.' — Suhandano 1994, p. 67, (104b)

(50) aku mangan-aké iwak menyang kucing — MALANG JAVANESE
1SG ACT.eat-AKE fish towards cat△
'I fed the fish to the cat.' — Hemmings 2013, p. 170, (10a)

Direct objects in Javanese receive no case or prepositional marking. They are placed immediately after the verb, which is a special position where only direct objects appear. Noun phrases that are placed after direct objects are preceded by prepositions and called "oblique" in Suhandano (1994, 35).

Although the same verbal suffix is used in both (49) and (50), the marking patterns are different. In (49), the causee (*dog*) is the direct object, and the affectee (*Bambang*) is oblique. In (50), the *cat* is the agent of the *eating*, i.e., a causee. It is placed in the position of an oblique and preceded by a preposition. The affectee (*fish*) is the direct object. Note that, according to source papers, examples (49) and (50) are taken from different varieties of Javanese.

Example (49) from Suhandano (1994, 67) has been cited in many other papers, including Dixon (2000). However, to my knowledge, no other source gives any other example of a causative derived from a transitive **activity** verb. Qomariana (2016) cites an example from Malang Javanese with the verb 'open', but the verb has a different suffix (*-no*, not attested in standard Javanese), and the sentence has a strong applicative meaning. Example (50) from Hemmings (2013, 170) has a verb of ingestion as its base. However, Hemmings (2013, 174) notes that the suffix *-ake* in this sentence does not increase the valency but preserves it. Ingestive verbs are known for their unusual behavior in causative constructions. Some evidence of that has already been shown in Bashkir (Section 6.1.1.4); see also the discussion about Lubukusu (Section 6.2.1.2.5).

The difference in verb classes is vital to the discussion of causatives that can be related to ditransitives. As shown in Section 6.1.1.4, in causatives based on verbs of physical or abstract transfer (including verbs of ingestion), the causee exhibits some properties of a prototypical recipient. Given that the verbal markers for both causative and ditransitive constructions are identical, one might question whether Javanese causative-of-a-transitive constructions exist at all. Or they are merely ditransitives and thus cannot be *related* to them.

6.1.2.2.2 Possible causative interpretation Let us nevertheless assume that (49) and (50) are causative constructions. The marking they exhibit resembles a pure ditransitive construction illustrated in (51) built with the same suffix -AKE.

(51) Ali m-(w)eneh-ake dolanan iku marang Zahra — Javanese
Ali ACT-give-AKE toy DEF to Zahra
'Ali gave the toy to Zahra.' — Sofwan 2010, p. 66, (70a)

In this ditransitive construction, the theme stands in the syntactic position of the direct object, and the recipient is expressed in the second postverbal slot, preceded by the preposition *marang*.

	Causer	Causee	Affectee
(49)	Sri	dog-DEF	*marang Bambang*
(51)	Ali	toy DEF	*marang Zahra*
	Actor	Theme	Recipient

Table 13: Relation of type **Affectee like Recipient** between (51) and (49) in Javanese

For the relation between (51) and (49) to qualify as **Affectee like Recipient**, it is necessary for both constructions to have the same number of core arguments. As for prototypical *give*-like constructions, Suhandano (1994, 39) and Sofwan (2010, 66) agree that oblique objects in them belong to the core. This conclusion relies on the fact that PPs in sentences like (51) cannot be omitted. Two-argument sentences with *give*-verbs are reported ungrammatical, which is a clear indication of three-argument cores in sentences like (51).

On the other hand, in constructions that do not involve a transfer of possession, the status of oblique objects is not totally clear. Remember that Hemmings (2013, 174) analyzes (50) as a sentence without the valency-increase. Thus, Hemmings (2013, 174) translates the verb *mangan-aké* as 'give as food'. The theme (*fish* in (50)) is a well-expected object of this verb.

When discussing applicative usages of the suffix -AKE, Sofwan (2010, 66) and Hemmings (2013, 168–169) agree that no three-argument predicate is built. The applicative is believed to replace the former direct object with a new one without increasing the valency of the predicate.

Causative constructions with the suffix -AKE can resemble either *give*-like ditransitive constructions and have three arguments or applicative constructions with -AKE and thus have two arguments. Since morphological causatives of transitive verbs are avoided (Malihah 2014), the impossibility of having three arguments in Javanese outside of *give*-like constructions is more plausible.

The lack of examples like (49) does not allow us to postulate the presence of the pattern **Affectee like Recipient** in Javanese. Thus, the assumption at the beginning of this section does not hold.

6.1.2.2.3 Type Causee like Recipient in Javanese For the sake of completeness, one must mention that Javanese has another ditransitive construction (52). The verb is marked with the suffix *-i*, and both non-subject participants are realized as bare NPs.

(52) Ali m-(w)eneh-i Zahra dolanan iku — JAVANESE
Ali ACT-give-I Zahra toy DEF△
'Ali gave Zahra the toy.'□ — Sofwan 2010, p. 66, (71a)

Suhandano (1994, 63) and Hemmings (2013) claim that the suffix *-i* can also build causatives, see (53)[19]. The corpus study by Malihah (2014, 65), however, does not attest any causative-of-a-transitive with the suffix *-i*.

(53) aku mangan-i kucing iwak — MALANG JAVANESE
1SG ACT.eat-I cat fish△
'I fed the cat fish.' — Hemmings 2013, p. 169, (6a)

Once paralleled, these two constructions look identical. Further studies in Javanese applicatives would show whether this pattern also qualifies as a causative-applicative syncretism or is an instance of the **Causee like Recipient** type.

6.1.2.3 Summary

In this section, I have hypothesized the existence of two other causative strategies that can be relatable to ditransitive constructions: **Causee like Theme** and **Affectee like Recipient**. After a careful look into ditransitive alignments, I offered a hypothesis on why these strategies are very infrequent. Afterward, I questioned the existence of morphological causatives derived from transitive verbs in modern standard Javanese and discarded their possible classification as type **Affectee like Recipient**. I also provided some data from another variety of Javanese that might have causatives of type **Causee like Recipient**. However, the syncretism of the verbal markers in Javanese leaves the question of whether these constructions count as causatives or applicatives open.

6.1.3 Causee like Instrument

Instrumental constructions, i.e., clauses with prototypical instruments like *John broke the window with a hammer*, are believed by Kemmer and Verhagen (1994) to be the second relatable construction for causatives (together with ditransitives). Section 6.1.1.5 showed that many languages use the same marking for prototypical instruments and demoted agents in passive (e.g., Kannada). It also showed a language that marks causees like demoted agents but not like instruments (Bashkir).

This section showcases another strategy: when causees are marked like instruments but not like demoted agents. The illustrative language of that kind is Finnish (Uralic > Finnic).

6.1.3.1 Data: Finnish

In Finnish, causees are typically marked with the suffix *-lla* 'ADESS', see (54), which is different from the marker used for demoted agents in passive, see (55).

(54) Mikko kuiva-tta-a pyykki-nsä naapuri-lla-an — FINNISH
Mikko dry-CAUS-3SG laundry-POSS.3 neighbour-ADESS-POSS.3
'Mikko has his neighbour dry his laundry.' — Nelson 1999, p. 148, (10b)

[19] However, Nurhayani (2014, 59) does not support this claim.

(55) a. Satu tapa-tt-i etana-n Diane-lla. FINNISH
Satu.NOM kill-CAUS-PST.3SG slug-ACC Diane-ADESS
'Satu had Diane kill the slug.'

b. *Etana tape-ttiin Diane-lla.
slug.NOM kill-PASS.PST Diane-ADESS.
*'The slugs were killed by Diane.' Manninen and Nelson 2004, p. 222, (20)

According to Karlsson (1999, 115), the *-lla* marker, which is traditionally glossed as ADESS, "often means location 'on top of' or 'near', 'owner', or 'instrument' by means of which an action is performed". Examples like (56) prove that the *-lla* suffix is used for marking prototypical instruments.

(56) Hän kirjoit-taa kynä-llä FINNISH
3SG write-PRS.3SG pen-ADESS
'He writes with a pen.' Karlsson 1999, p. 116

It is not uncommon for Finnish locative cases to have non-locative usages. For instance, Vainikka (2015, 130, fn. 2), focusing mainly on elative, reports non-canonical usage of ablative and also adessive: "Adessive case is the case of some instrumental phrases (*sekoittaa kauha**lla*** 'to mix with a spoon(ADE)') and the case of the 'agentive' NP in causative constructions (as pointed out to me by U. Nikanne)" (emphasis mine – VG)). Toivainen (2015, 113) goes further in stating that "the adessive is counted as an instrumental case in its instrumental use, otherwise it is a locative case". Pushing the perception of adessive as instrumental to the extreme, Pylkkänen (1997) even glosses the *-lla* suffix as INS, see (57) with original glossing kept.

(57) Kalle rakenn-utta-a talo-n Peka-lla. FINNISH
Kalle.NOM build-CAUS-3SG house-ACC Pekka-INS
'Kalle has Pekka build a house.' Pylkkänen 1997, p. 3, (6)

This notation is especially useful when discussing causative constructions since purely locative cases are very unlikely to appear on core arguments. In my work, I gloss the suffix *-lla* as ADESS as in traditional grammars, acknowledging that this primarily locative case can also be used in an instrumental function. In turn, I call the strategy of argument marking in Finnish constructions with a morphological causative **Causee like Instrument**.

Locative vs. non-locative usages of Finnish cases can be approached from the perspective of predicative vs. non-predicative adpositions offered in Van Valin (2005, 21–23). According to it, "predicative adpositions function like predicates in that they contribute substantive semantic information to the clause in which they occur, both in terms of their own meaning and the meaning of the argument that they license" (Van Valin 2005, 21). A typical example of a predicative adposition would be a locative adposition in its locative usage: *Kim put the book* ***in the box*** (example by Van Valin (2005, 22)). Predicative adpositions introduce adjuncts. In contrast, non-predicative adpositions "do not add any substantive semantic information to the clause and do not license the argument they mark. Rather, their argument is licensed by the predicate, i.e., it is a core argument" (Van Valin 2005, 21–22). An example of an English non-predicative adposition is *to* in its non-locative usage, as in *Kim gave the book* ***to Pat***.

Morphological cases can also be predicative and non-predicative. Hence, Finnish locative cases in their locative usage can be perceived as predicative and thus introduce adjuncts. In contrast, Finnish locative cases in their non-locative usage – namely, ADESS in its instrumental function – could be regarded as non-predicative. Thus, it is suspected of introducing core arguments.

6.1.3.2 A prior non-RRG analysis

Finnish causatives received an event-based analysis in Pylkkänen (1997), being an elaboration of Parsons (1990) (who, in turn, follows the general neo-Davidsonian approach). According to it, the logical structure for (57) looks like (58).

(58) (∃e)[Agent(e,Kalle) & Theme(e, Pekka) & (∃e')[Build(e') & Agent(e',Pekka) & Theme(e',house) & CAUSE(e,e')]]

This structure basically tells the following: (i) *Kalle* and *Pekka* are participants of the same (not named) event, *Kalle* being the Agent and *Pekka* being the Theme; (ii) *Pekka* and *house* are both participants of the event of *building*, *Pekka* being the Agent and *house* being the Theme; (iii) there is a causative relation between the first not named event and the event of *building*.

The author notices that "detailed representations of event structure often involve more event participants than what seemingly are present in the syntax" (Pylkkänen 1997, 14). To ensure correct linking between the logical representation and the syntactic structure, Pylkkänen (1997) uses two constraints. The first one, called *uniqueness*, is formulated in (59).

(59) Uniqueness:
(i) For all eventualities E_n in a lexical representation L, every participant of E_n must have a unique syntactic function.
(ii) No two syntactic functions can be linked to any one event participant of L.
Pylkkänen (1997, p. 8, (22))

The second constraint, called *identity*, states that some participants in the logical structure are linked to the same syntactic constituent. Translating the representation of identity offered in Pylkkänen (1997, p. 9, (25)) into words, one gets the following linking rule:

(60) Two participants of the event structure can be linked to the same entity of the syntactic structure if
(i) they are participants of different events
(ii) the rank of the second participant is one step lower than the rank of the first one.

The mentioned ranks are determined based on the *prominence hierarchy* formulated earlier in the paper (Pylkkänen 1997, 7) as the interaction of the two hierarchies shown in (61).

(61) a. Thematic Hierachy: Agent/Experiencer > Other > Theme

b. Event Hierarchy: CAUSE(E1,E2) $\rightarrow E1_{\theta H} > E2_{\theta H}$

All these principles ensure the correct linking so that the Agent of the causing event (e) in (58) *Kalle* is linked to the syntactic subject of (57), the Theme of the cased event (e') *house* is linked to the syntactic object, and all the instances of the third participant *Pekka* are linked to the single third argument of (57), which is marked with ADESS (or INS in Pylkkänen's terms).

6.1.3.3 An RRG analysis

Nevertheless, the analysis suggested in Pylkkänen (1997) does not tell much about the specificity of the Finnish causative strategy. Why does the causee receive ADESS marking? Is it actually perceived as an instrument? Is it a core argument or an adjunct? These questions remain open.

A classical RRG argument would suggest that ADESS in its non-locative usage is a non-predicative case and thus that *Pekka-lla* 'Pekka-ADESS' in (57) is a core argument. The instrumental-like case is assigned to it by the general rule (see Section 2.5.3 for details): both *Kalle* and *Pekka-lla* are effectors, but *Kalle* outranks *Pekka-lla* on the Actor-Undergoer Hierarchy. Thus, *Kalle* receives NOM and *Pekka-lla* receives ADESS, which, in Finnish, is syncretic with INS (or is the INS).

However, there are several hints indicating that 'Pekka-lla' is in the periphery. Initially a locative case, even in a non-locative usage, ADESS still bears most of its locative-like syntactic characteristics. Although there are elaborate systems of locative case assignment (Nikanne 1997) and also some other works studying in much detail non-locative Finnish cases (Kiparsky 2001), the non-locative usage of the locative case has not received much attention. Nikanne (2015) discusses how locative cases are assigned and comes to the conclusion that "according to the distribution, binding, and predication facts, the locative cases and other semantic cases seem to form PPs where the NP in the locative cases is embedded by some head. <…> The most plausible possibility seems to be that the semantic cases are assigned by empty Ps, each of which corresponds to exactly one locative case." (Nikanne 2015, 86; see an earlier version in Nikanne 1989). In

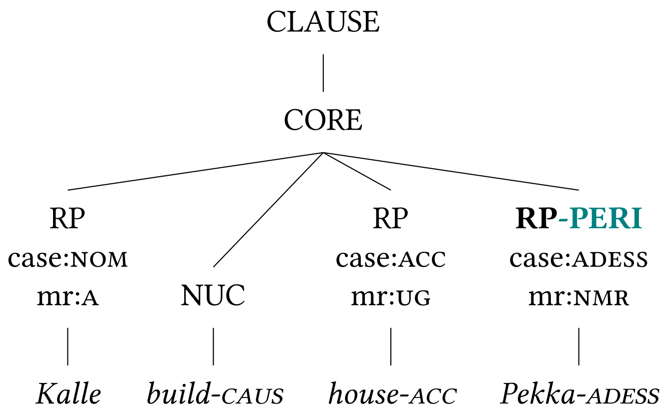


Fig. 11. The syntactic representation of Finnish causative-of-transitive (57) as a two-argument core

Nikanne (1989), the author suggests that non-locative usages of locative cases (including instrumental) are just interpretations. In Nikanne (2015, 81–82), speaking of possessive constructions where ADESS marks the possessor, the author notes that sometimes these sentences can be interpreted as either locative or possessive. However, sometimes the locative (not the possessive!) interpretation is not possible. Given that, Nikanne (2015, 82) concludes "that the possessive construction is lexicalized somehow", making no attempt to re-interpret the adjunct status of the ADESS possessor.

Given this, it seems plausible to postulate the structure of Finnish morphological causatives based on transitive verbs as a two-argument core, as shown in Fig. 11.

6.1.3.4 Discussion

The next question is the place of Finnish constructions compared to constructions in other languages.

If one compares the three positions in question where effectors can appear – namely, the Agent (of an active clause), the Demoted agent, and the Instrument – one can make the following syntactic and semantic observations, see Tab. 14. In terms of syntactic coding, the Agent differs from the other two participants, as it is most often marked as the PSA (in accusative constructions). Both Demoted agents and Instruments are usually non-PSA arguments or can even be adjuncts. In terms of semantic agentivity, Demoted agents are equally high as Agents of active clauses. The fact of syntactic demotion does not entail lower agentivity. In contrast, Instruments are less agentive.

	Agent	Demoted agent	Instrument
syntactic coding	PSA	non-PSA	non-PSA
semantic agentivity	high	high	low

Table 14: Agent, Demoted agent, and Instrument compared

	Marking		
	Causee	DemAg	Instrument
Finnish	ADESS	not ADESS	ADESS
Bashkir	ABL	ABL	not ABL
Kalmyk	INS	INS	INS

Table 15: Marking of causees, demoted agents and instruments in Finnish, Bashkir and Kalmyk

The distribution of the two features listed in Tab. 14 can give a hint why some languages use the same marker for both Demoted agents and Instruments, while some other languages do not (examples of different situations given in Tab. 15). If the difference in semantic agentivity is crucial for a language, different ways of marking are used for demoted agents and instruments (like Finnish and Bashkir in Tab. 15). If this difference is less relevant, a syncretic marker appears on both of these (e.g., Kalmyk in Tab. 15, or also Kannada discussed in Section 6.1.1.1.3).

When it comes to the coding of the causee, in languages like Kalmyk and Kannada, it receives the same syncretic marker. It is thus impossible to tell whether the causee is marked like the demoted agent or the instrument. Nevertheless, this difference is of lesser importance in these languages, so calling this strategy syncretic without adding further precision might be the analysis most adequate to the language reality. In contrast, languages like Bashkir and Finnish distinguish demoted agents and instruments and relate causees to only one of them. Thus, for these languages, it makes sense to distinguish between the strategies **Causee like Demoted Agent** and **Causee like Instrument**. Hence, once a quantitative typological study aims to determine the frequency of each causative strategy, it must be sensitive to the three strategies mentioned above. Presumably, the syncretic strategy will appear more frequently than the non-syncretic ones.

6.1.3.5 Summary

So far, Finnish causative construction has been known as a very rare representative with a locative causee. In this chapter, I have re-analyzed it as relatable to the construction with a prototypical instrument and questioned whether it is that rare. In addition, I drew attention to the difference between three strategies: **Causee like Demoted Agent**, **Causee like Instrument**, and a syncretic one, where all the three participants in question are marked identically.

6.1.4 Causee like Agent of a Converbial Clause

Continuing the line of the strategy **Causee like Demoted Agent**, one needs to pay special attention to Nivkh. (This language is known to have a very special causative strategy, where the agentivity of the causee plays the central role. In prior studies, including Dixon (2000), Nivkh was claimed to have a special causative construction that could not be related to any other. In this section, I suggest relating Nivkh causatives to certain converbial clauses and discuss how Nivkh data contribute to understanding the argument structure of causatives in general.

6.1.4.1 Data: Nivkh

In Nivkh, the causee receives the case marker -*aχ* in constructions with both intransitive (62) and transitive (63), (64) bases.

(62) ytyk p‘-oɣla-aχ vi-gu-d — Nivkh
father poss-child-dat/acc go-caus-fin
'The father sent his son (away).' — Nedjalkov and Otaina 2012, p. 51, (86b)

(63) ytik n‘-aχ mos amla-gu-d — Nivkh
grandmother 1sg-dat/acc berry.pudding taste-caus-fin
'The grandmother let me eat the berry pudding.'
Nedyalkov, Otaina, and Xolodovič 1969, p. 192

(64) ytyk p‘-oɣla-aχ pitɣy ama-gu-d — Nivkh
father poss-child-dat/acc book see-caus-fin
'The father showed his son the book.' — Nedyalkov, Otaina, and Xolodovič 1969, p. 192

Earlier studies (Krejnovič 1934; Panfilov 1962) report that this marker appears exclusively in causative constructions. This has raised the assumption (shared by Dixon (2000) among others) that in Nivkh, causative cannot be related to any other syntactic construction. Kemmer and Verhagen (1994) argue against the existence of such a strategy when their approach (from which mine is derived, see Section 5.4) is applied.

> Fourth, our Schema approach suggests an explanation for why languages that have a separate case marking to express the causee are rare or nonexistent. This is a fact that (as far as I am aware) has not been previously noted. If, as we claim, clause Schemas with simple predicates are elaborated as Schemas with causative predicates, it makes sense that the case markings found in simpler structures would be those used in the more elaborated constructions. In fact, we might make the stronger prediction that the use of a case marker for a given participant in a causative clause requires there to be a corresponding simple clause structure with a similarly marked participant, relatable to the first participant both in terms of the overall participant configurations of the clause structures and the semantic specifications of the case marker itself. — Kemmer and Verhagen (1994, 147)

Speaking especially of Nivkh (Gilyak), Kemmer and Verhagen (1994) remains suspicious: "It is unclear whether this marker can be considered a "causee case marker", however, since the language does not have a case-marking system in the usual sense" (Kemmer and Verhagen 1994, 152).

In this section, I am presenting some evidence from recent studies that might suggest relating Nivkh causee marking to some type of other construction.

Note that the glossing of the marker -*aχ* varies from source to source. It is called "agentive case" in Nedjalkov and Otaina (2012), "dative-accusative" in Gruzdeva (1998) as well as "causative accusative", "second accusative" in some earlier studies (see Nedjalkov and Otaina (2012, 51) for a list of names and references). I decided to use the label 'DAT/ACC' (as in Gruzdeva (1998)) because most of the examples are taken from this source. Nevertheless, it is necessary to discuss the characteristics of this suffix in more detail, as the label does not adequately reflect them.

Importantly, Nedyalkov, Otaina, and Xolodovič (1969, 195) report that the -*aχ* marker can appear only on animate nouns. Panfilov (1962, 131) restricts the nouns that can take this marker even more: "animate entities that can be objects of causation" (i. e, that can be causees – VG). However, even animate nouns do not always receive explicit marking with -*aχ*. Nedyalkov, Otaina, and Xolodovič (1969, 194) claim that in case a sentence already has two unmarked noun phrases (subject and direct object), the marker -*aχ* cannot be omitted. In reality, its distribution depends on other predicative elements (converbs) and reference phrases (nouns, pronouns, agreement morphemes) in the clause, and this is a separate topic in Nivkh studies. Currently, I would limit myself to clear cases like (63), (64).

Inanimate nouns are, anyway, extremely rare causees of causative-of-transitives. According to Nedyalkov, Otaina, and Xolodovič (1969), inanimate nouns appear in this function if they are "activated due to former activity of a human being" (Nedyalkov, Otaina, and Xolodovič 1969, 194). The example of such a case suggested in Nedyalkov, Otaina, and Xolodovič (1969, 194) would be a *shot bullet*, which, although inanimate *per se*, exists in its shot state only because of human activity and can be thus perceived as a result of an action by a highly agentive and presumably volitional subject. In this situation, the noun *bullet* remains unmarked.

6.1.4.2 Relatable constructions

Recent works, namely, Nedjalkov and Otaina (2012) and Gruzdeva (1998) list some usages of the marker -*aχ* outside of causative constructions. One is that it marks the subject of a converb built with the suffix -*gu*-, which, according to Nedjalkov and Otaina (2012, 51), has no more causative meaning in this case. Gruzdeva (1998) glosses it as a purpose converb, see (65).

(65) Nivkh

hoĝat	n’yṇ	čyṇ	ṇyṇ-d’-ra	čyṇ-aχ	p‘-ro-guin
then	we	you	look.for-FIN-PART:PRED	you-DAT/ACC	REFL-help-CVB:PURP

‘We were looking for you in order that you help [us].’ Gruzdeva 1998, p. 52, (127)

Lit: ‘We were looking for you in order to have you help us.’ Rainer Osswald, p.c.

One can see that (65) can still be read as a causative, cf. the translation suggested by Rainer Osswald (personal communication). Thus, (65) is not sufficient proof that the marker *-aχ* occurs in non-causative constructions.

Another domain where the marker *-aχ* is found is the reported speech, where it marks the semantic subject of a "re-telling converb" *-vu-r/-vu-t* (Gruzdeva 1998, 57), see (66). Crucially, the subject of the converb is never the same as the subject of the finite verb, irrespective of its marking. Consider (67) and the indices in translation: the one who comes is not the one who says.

(66) Nivkh

if	imṇ-aχ	als	p‘e-ny-vur	it-t’
he	they-DAT/ACC	berry	pick-FUT-CVB:RTL	say-FIN

‘He said [that] they would pick berries.’ Gruzdeva 1998, p. 57, (148)

(67) Nivkh

a.

ytyk	if	p‘ry-vur	it-t’
father	3SG	come-CVB:RTL	say-FIN

b.

ytyk	j-aχ	p‘ry-vur	it-t’
father	3SG-DAT/ACC	come-CVB:RTL	say-FIN△

Both: ‘The father$_i$ says/said that he$_j$ (*he$_i$) came.’□

Nedjalkov and Otaina 2012, p. 127, (495ab)

Nedjalkov and Otaina (2012, 52) mention that "apparently, *-vu-* traces its origin back to *-gu-*". So, there is a clear tendency: the *-aχ* marker appears only in constructions, where the verbal marker *-gu-* is found or can be reconstructed. However, the use of this verbal marker is broader than just causative constructions. The fact that the *-aχ* marker is compatible with a single verbal form raises concerns about whether it can count as a case at all.

This issue remaining far beyond the scope of the present study, the important thing is that the *-aχ* marker is clearly intended to encode a demoted subject. In reported speech, the subject of the embedded predicate retains all its semantic properties but syntactically cannot be realized as the subject of the clause. Therefore, it has to receive a different marking, and in Nivkh, it is the *-aχ* marker. Likewise, in causative constructions based on transitive verbs, the former actor of the base verbs becomes the causee but retains its semantic agentive properties. It is thus demoted to the same position as the agent in a reported clause.

Identical marking of former agents of reported clauses and of the causees leads to the conclusion that they also share some important features: high agentivity, animacy, etc. This claim is supported by prior analyses that focus on the semantics of causatives, while disregarding their syntactic resemblance to other constructions. The fact that the -*aχ* marker combines exclusively with animate nouns also speaks for the high agentivity of the participant it attaches to. Animacy is not the sole criterion of agentivity, but a strong correlation has always been observed between the two.

An interesting observation confirming the highly agentive nature of -*aχ*-marked participants has been made by Kaneko (2006). Examining sentences with finite verbs having the marker -*gu*-, the author notices that even the animate causees do not necessarily receive the -*aχ* marker. According to Kaneko (2006, 8), for the causee to be marked, it has to be an "active actor". Yet tautological, this description is very precise. Some animate causees can, in fact, lack agentive properties and, for that reason, receive no additional marking. Kaneko (2006) concludes with the assumption that the -*gu*- suffix lacks causative meaning in sentences where all participants lack the -*aχ* marker. In other words, the author claims the existence of two distinct three-argument constructions with distinct marking strategies, which I fully agree with. Not only does this claim confirm other observations related to the agentivity of the -*aχ*-marked causees, but it also sheds light on the distributional and semantical properties of the marker itself.

As a side note, in this light, the term "agentive case" used for the -*aχ* marker by Nedjalkov and Otaina (2012) seems to be a rather successful choice, the limitations on the usage of this case aside.

6.1.4.3 The suggested RRG analysis

Based on previous observations, one can conclude that the causee in Nivkh causative constructions is perceived not as an undergoer but as a second-high actor. Although the causatives of intransitive bases like (62) are outside of the present work's focus, I assume that, in RRG terms, they are, in fact, bivalent one-macrorole sentences lacking the Undergoer. In three-argument constructions, the causee is the non-macrorole core argument. According to the traditional RRG analysis, it could have received the instrumental case, but in Nivkh, it does not. Thus, Nivkh clearly has a special constructional schema for causatives. In this respect, Nivkh is similar to Bashkir in marking causees with a case distinct from that of recipients and instruments. However, as Nivkh lacks finite passive clauses, the option for marking the causee as a demoted agent in passive is unavailable. One can say that the -*aχ* suffix marks demoted agents in clauses with the -*gu*- verbal marker, be they finite or converbial. In this perspective, Nivkh can be regarded as a subtype of strategy **Causee like Demoted Agent 2**. The constructional schema for Nivkh causatives is shown in Tab. 16. The relatability to the reported speech is questionable since it is not clear whether the verbal marker -*gu*- is different in causatives and converbs of re-telling or is some kind of a syncretic marker that alone triggers the -*aχ* marking.

Strategy:	Causee like Demoted Agent 2
Relatable to:	reported speech (?)
Argument encoding:	$E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{tr}\ U_k^{tr}\ :\ demA_j^{rep}$
Macrorole assignment:	$E_a^{caus}\ C_{nmr}^{caus}\ A_{ug}^{caus}$

Table 16: Constructional Schema for Nivkh causatives

The last concern in this chapter is about relating the semantic structures of Nivkh causatives to the semantic structures of the same verbs in other languages. Nivkh examples cited in this section are built with verbs of ingestion (63) and non-physical transfer (64). In languages having several constructions with a morphological causative, these verb classes tend to require a reading where the causee is closer to the recipient than to the agent.

The lack of data does not allow us to determine whether the causative strategy with -*aχ* is special to these verb classes or is determined by a different parameter of variation. Since the -*aχ* suffix appears in constructions with causatives derived from intransitive bases, it seems plausible that the verbal class does not influence the selection of the causative construction in Nivkh. In this respect, Nivkh differs from Bashkir (which has several causative strategies) and is similar to, for example, Finnish.

6.1.4.4 Summary

The section has shown that Nivkh indeed has a peculiar strategy of argument marking in causative constructions. However, the method relating causatives to other constructions permits regarding it as a variant of a more widespread strategy **Causee like Demoted Agent**. Nevertheless, the co-occurrence of the case-like suffix -*aχ* and the verbal marker -*gu*- questions the case status of -*aχ*. Further studies in Nivkh could perhaps offer a more precise description of its functions.

6.2 Strategies with doubling

6.2.1 Double objects

6.2.1.1 General notes

Bantu languages are known for their double-object constructions. Under double objects, one traditionally understands two identically marked NPs, both in the position of an object. In Bantu languages, this definition is usually applied to constructions like (68) having two postverbal NPs, as the subject usually precedes the verb and, in some languages, is also cross-referenced on the verb (consider the C1.SBJ morpheme in all examples in (68)).

(68) a. KINYARWANDA

Karemera	a-ra-tek-er-a	umw-umbati	umw-ana	we
Karemera	C1.SBJ-PRS-cook-BEN-IPFV	C3-cassava	C1-child	C1.his

'Karemera is cooking a cassava for his child.' Jerro 2017, p. 755, (6b)

b.

Karemera	a-ra-kubit-ish-a	uru-kuta	in-koni
Karemera	C1.SBJ-PRS-hit-CAUS-IPFV	C10-wall	C9-stick△

'Karemera is hitting the wall with a stick.' Jerro 2017, p. 755, (6c)

c.

Umw-arimu	y-a-ndik-ish-ije	umw-ana	in-kuru
C1-teacher	C1.SBJ-PST-write-CAUS-PFV	C1-child	C9-story

'The teacher made the child write the story.' Jerro 2017, p. 759, (16)

The crucial property of these constructions is that both non-agent NPs are marked in the same way, which is also identical to the marking of the non-agent participant in a prototypical transitive clause (hence the term *double object*), cf. (69).

(69) KINYARWANDA

Habimana	y-a-men-a	igi-kombe
Habimana	C1.SBJ-PST-break-IPFV	C7-cup

'Habimana broke the cup.' Jerro 2017, p. 753, (3)

One of the complications with Bantu double object constructions is the lack of overt NP marking. Scholars have searched for differences between object-like Bantu NPs for years. Since syntagmatic contrast (segmental marking, ordering constraints) was not found, paradigmatic variation was explored. Finally, it has been discovered that in some Bantu languages, one of the postverbal NPs has more object-like properties than the other, and thus the two NPs are not identical. These languages were called *asymmetrical*. In other languages, however, this difference is not attested (hence the term *symmetrical*). Some more details on this debate are offered in Section 6.2.1.2, along with recent additions to it from the HPSG perspective. Later, I offer an RRG view on this matter and suggest constructional schemas for Bantu causative constructions using updated notions of the object (a)symmetry.

Analyses seeking criteria for the object (a)symmetry imply that there should be some difference between NPs. In other words, they imply that the locus of variation is on the *dependent* constituents. In the meantime, Bantu verbs have some prefixes that agree with NPs in class. This could characterize Bantu languages as (partially) head-marking. However, the nature of the verbal prefixes calls this analysis into question. Prior debates and my reflections on this topic are offered in Section 6.2.1.3.

6.2.1.2 Challenge 1: Object (a)symmetry

6.2.1.2.1 Key notions Since Bresnan and Moshi (1990), it has been suggested to divide Bantu languages in *symmetrical* and *asymmetrical*. Later studies have shown that these terms are better applied to constructions since both symmetrical and asymmetrical patterns can be encountered within the same language (see Section 6.2.1.2.5).

In asymmetrical constructions, identically looking object-like NPs have different syntactic properties, which can be discovered through syntactic tests. This is what Kozinsky and Polinsky (1993) call 'covert coding conflict resolution'. For example, in asymmetrical patterns, only one NP can become the subject of the passive clause. In a broader perspective, three-argument constructions without a coding conflict (e.g., constructions with an ACC direct object and a DAT indirect object) also can be regarded as asymmetrical since only one NP demonstrates the full range (or the largest number) of object-like properties. This basic assumption of asymmetry being universal is questioned by Ackerman, Malouf, and Moore (2017) (see below).

Symmetrical languages (or rather constructions) allow any of the two non-subject participants of a three-argument construction to demonstrate the full (or equal) range of object-like properties. Example (70) shows a causative construction with two objects, and (71) proves that any of them can become the subject of a passive construction (which is indicated by the class marker on the verb after the SBJ morpheme).

(70) Wafula a-nyw-esy-a Wekesa ka-ma-lwa — LUBUKUSU
Wafula SBJ.C1.TNS-drink-CAUS-FV Wekesa C6-C6-beer
'Wafula made Wekesa drink beer.' — Baker, Safir, and Sikuku 2012, p. 54, (1)

(71) a. Wekesa a-nyw-esy-ebw-a ka-ma-lwa — LUBUKUSU
Wekesa SBJ.C1.TNS-drink-CAUS-PASS-FV C6-C6-beer
'Wekesa was made to drink beer.'

b. Ka-ma-lwa ka-nyw-esy-ebw-a Wekesa
C6-C6-beer SBJ.C6.TNS-drink-CAUS-PASS-FV Wekesa
'Beer was made to be drunk by Wekesa.' — Baker, Safir, and Sikuku 2012, p. 54, (2)

Hence, Lubukusu can be called a symmetrical language since it allows symmetrical patterns like (70). The verb *-nyw-* 'drink' can be called a symmetrical predicate since it functions in symmetrical three-argument constructions.

To compare, the asymmetrical situation suggests that one of the examples in (71) would be ungrammatical. Interestingly, this type of construction is also found in Lubukusu, see (72)-(73).

(72) O-mw-ekesi ∅-a-p-isy-a o-mu-khangarani li-sisi Lubukusu
c1-c1-teacher c1.sbj-pst-hit-caus-fv c1-c1-warrior c5-wall
'The teacher made the warrior hit the wall.' Jerro 2019, p. 220, (1)

(73) a. Lubukusu

O-mu-khangarani ∅-a-p-isy-ibw-a li-sisi ne o-mw-ekesi
c1-c1-warrior c1.sbj-pst-hit-caus-pass-fv c5-wall by c1-c1-teacher
'The warrior was made to hit the wall by the teacher.'

b. ??Li-sisi ly-a-p-isy-ibw-a o-mu-khangarani ne o-mw-ekesi
c5-wall c5.sbj-pst-hit-caus-pass-fv c1-c1-warrior by c1-c1-teacher
'The wall was made to be hit by the warrior by the teacher.'

Jerro 2019, p. 220, (2)

Thus, the verb *p-* 'hit' in Lubukusu, as well as the pattern (72), can be called asymmetrical.

The very phenomenon of object symmetry sparked an animated debate aimed at explaining its causes. The present section does not aim for that. Instead, it evaluates whether causative constructions in either symmetrical or asymmetrical Bantu languages repeat the ditransitive or any other pattern found elsewhere in the same language.

6.2.1.2.2 A prior HPSG analysis Among other works about object (a)symmetry in Bantu, I find the paper by Ackerman, Malouf, and Moore (2017) the most relevant for the current study. After offering a sound summary of prior research, the authors mention that most of the previous explanations of symmetry treated **a**symmetry "as a default universal grammatical pattern, with departures from it requiring modifications of otherwise more simple assumptions" (Ackerman, Malouf, and Moore 2017, 4–5). In other words, the authors suggest reconsidering the idea of symmetry as a deviation from asymmetry. The novel approach proposed by Ackerman, Malouf, and Moore (2017) aims to provide an analysis that accounts for both asymmetrical and symmetrical patterns as realizations of a single, more general principle.

For doing that, Ackerman, Malouf, and Moore (2017) suggest departing from the Argument Selection Principle, initially formulated in Dowty (1991), see (74).

(74) Argument Selection Principle (adapted from Dowty 1991, 576)

a. In predicates with grammatical subject and object, the argument for which the predicate entails the greatest number of proto-agent properties will be lexicalized as the first item on the predicate's ARG-ST.

b. The argument for which the predicate entails the greatest number of proto-patient properties will be lexicalized as the second item on the predicate's

ARG-ST. Ackerman, Malouf, and Moore (2017, p. 32, (51))

The main contribution by Ackerman, Malouf, and Moore (2017) is the suggestion "that in a symmetric language, only the first clause of the Argument Selection Principle (74a) is active". When formulating the Argument Selection Principle for Moro (a Heibanic (Kordofanian) language that has many similarities with Bantu languages), the authors claim that "the remaining [non-subject – VG] arguments are unordered in the predicate's ARG-ST" (Ackerman, Malouf, and Moore 2017, 33). As a result, an ARG-ST with three arguments, the first of which is fixed, yields two fully determined linear realizations: 1-2-3 and 1-3-2 (numbers being used here just as indices). Other properties of the constructions (including realization as an incorporated pronoun, see Ackerman, Malouf, and Moore 2017, 35–36) can be determined by combining this rule with others.

The ARG-ST level occupies a special place in HPSG structures.

> Crucially, the argument structure (ARG-ST) is distinct from both the valence features (VAL), which govern how a verb combines with its dependents syntactically, and from the semantic features (SEM), which determine the thematic roles assigned to each argument.
>
> Ackerman, Malouf, and Moore (2017, 31)

Being a kind of interface between syntax (VAL) and semantics (SEM), it somehow corresponds to the notion of macroroles in RRG. Indeed, in HPSG, the ARG-ST is a general level that determines the roles of the arguments. In RRG, macrorole is a general concept, and the macrorole structure of a verb at least partly determines the roles of the arguments. The ordering of arguments in the ARG-ST relies on different factors and determines their syntactic realization. Similarly, in RRG, the coding of an argument depends on its position on the Actor-Undergoer Hierarchy. Therefore, claims made in HPSG about ARG-ST can be translated to RRG in terms of macroroles.

6.2.1.2.3 An RRG analysis The claim about two arguments being "unordered in the ARG-ST" would thus translate in RRG like *no argument being assigned a macrorole*. In RRG, macrorole-bearing arguments prevail over non-macrorole arguments. If something is "unordered", it must then have equal macrorole status. Since there are only two macroroles and one is reserved for the subject, both objects cannot be equal, and both bear a macrorole. Hence, the only logical option is that neither of them bears any macrorole.

At first glance, this idea contradicts the basic principle of macrorole assignment in RRG (see (2.41a) from Van Valin (2005, 63) reproduced below as (75)):

(75) If a verb has two or more arguments in its logical structure, it will take two macroroles.

This violation is not surprising, though: it has been discussed with respect to various languages and phenomena and has been found to be common. It has been noted that some verbs demonstrate systematic discrepancies between the number of arguments and the number of macroroles they take (see Van Valin (2005, 63–64) and especially Tab. 2.5. on p. 64). For example, "multiple-argument activity verbs with a non-referential second argument, e.g., *beer* in *Kim drank beer*, never have an undergoer macrorole" (Van Valin 2005, 63). Here, the verb *drink* has two semantic arguments and two syntactic direct core arguments, but only one macrorole-bearing argument. In terms of grammar description, this ability of the verb can be either part of its own lexical entry or described as a systematic language-level alternation. Hence, verbs like *drink* in *Kim drank beer* and *-nyw-* 'drink' in (70) should receive additional lexical specifications concerning the number of macroroles they take. In languages that have exclusively symmetrical patterns, this specification can be introduced for all verbs as part of the grammar.

What is special about Bantu double object causative constructions is that the number of arguments they have is greater than the number of macroroles **by 2**. Most languages have a maximal difference between M-transitivity and S-transitivity equal to 1. Exceptions nevertheless exist. Icelandic trivalent verbs like (76) are analyzed in RRG as having one single macrorole, but three direct core arguments (Van Valin 2005, 118, Van Valin and LaPolla 1997, 352–361, Van Valin 1991).

(76)	Eg	skila-ð-i	henni	pening-un-um	Icelandic
	1sg.nom	return-pst-1sg	3fsg.dat	money-def-dat	

'I returned her the money.' Van Valin 2005, p. 118, (4.48a)

In Van Valin (2005, 118), Van Valin (1991), the authors note that constructions like (76) have often been approached as having "irregular" or "quirky" case assignment, to which complicated explanations have been needed. Opposite to that, the RRG analysis with two non-macrorole arguments claims one irregularity of the verb – the difference of 2 between the M-transitivity and the S-transitivity – from which all other properties of the constructions are derived following regular rules.

I believe that the same can be claimed for Bantu constructions like (70): at the syntactic level, they have three arguments, but at the macrorole level – only one[20]. The single macrorole-bearing argument is *Wafula*, the preverbal NP, the syntactic subject (= the psa). None of the participants *Wekesa* or *beer* is an Undergoer.

This analysis raises one important question about passives in Lubukusu. Traditionally, it is believed that only Undergoers can become syntactic subjects (psa in RRG terms)

[20]This claim implies that non-causative bivalent constructions like *Wekesa drinks beer* in Lubukusu also have a single macrorole-bearing argument, i.e., that *Wekesa* is an Actor, but *beer* is not an Undergoer. In this line, causative morphemes allow the sentence to have one more syntactic argument but do not operate on the macrorole level. Otherwise, if non-causative bivalent constructions are claimed to have two macrorle-bearing arguments, the analysis of the causative derivation would look disturbingly unusual: the S-transitivity would increase, but the M-transitivity would decrease.

in passive constructions. If there is no Undergoer in (70), why would a passive (and even two) be possible? In fact, existing analyses of Icelandic could also help with this issue. In Van Valin (2005, 119), following Roberts (1995), the author suggests that the passive rule for Icelandic contains the claim "non-Actor becomes PSA" instead of the traditional "Undergoer becomes PSA". This claim works on verbs with any valency, including non-causative bivalent, non-causative trivalent, and causative trivalent ones. Apparently, the same rule holds for Lubukusu (and, presumably, other languages featuring symmetrical double-object constructions). Moreover, it may be claimed that this rule often co-occurs in languages with greater discrepancies between the values of M-transitivity and S-transitivity.

6.2.1.2.4 Zulu: Data and analysis The analysis suggested in Section 6.2.1.2.1 can be unproblematically applied to fully and systematically symmetrical languages. According to Van der Wal (2018), a language of that kind is Zulu: "both objects behave symmetrically, whether they belong to a lexical ditransitive verb or a derived applicative or causative. This is illustrated for object marking in (77) and yields the same results for passivization" (Van der Wal 2018, 118). Although (77) features the results of the pronominalization test (in contrast to the passivization test used throughout this section), it clearly demonstrates that the pattern is symmetrical.

(77) a.

ULanga	u-phek-is-a	umama	ukudla	ZULU
C1A.Langa	C1SBJ-cook-CAUS-FV	C1A.mother	C15.food	

'Langa helps/makes mother cook food.'

b.

ULanga	u-m-phek-is-a	ukudla	(umama)
C1A.Langa	C1SBJ-C1OBJ-cook-CAUS-FV	C15.food	(C1A.mother)

'Langa helps/makes her cook food (mother).'

c.

ULanga	u-ku-phek-is-a	umama	(ukudla)
C1A.Langa	C1SBJ-C15OBJ-cook-CAUS-FV	C1A.mother	(C15.food)

'Langa makes mother cook it (the food).' Van der Wal 2018, p. 119, (8a-c)

Thus, for all verbs in Zulu, one must postulate M-transitivity = 1. This restriction applies at the level of the whole language grammar since all the verbs that can have multiple objects exhibit the symmetrical argument pattern.

The rule for pronominalization would then be "any non-Actor argument can become an object pronoun" (consider C1OBJ in (77b) and C15OBJ in (77c)). This rule is also language-wide since all verbs in Zulu exhibit pronominalization patterns like (77b)-(77c).

Given that, (77a) is analyzed as follows. The single macrorole-bearing argument in (77a)-(77c) is *Langa*, its macrorole is Actor. Moreover, this value does not change with changes in S-transitivity. The S-transitivity in (77a) is 3, and all the NPs are direct core

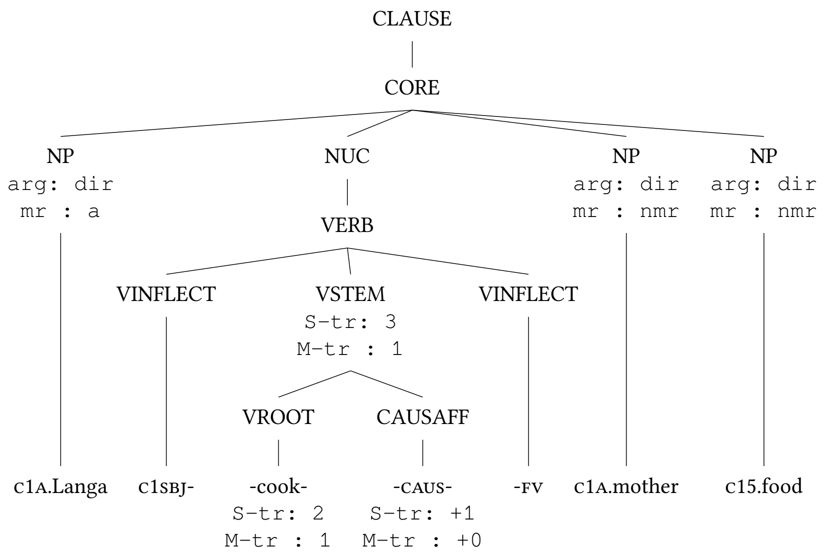


Fig. 12. The tree representation of a Zulu causative (77a)

arguments.

The tree resulting from the suggested analysis is shown in Fig. 12. Verbal morphemes are decorated with features `S-transitivity` and `M-transitivity`. The values of these features assigned to the verbal root combine with the values of these features assigned to the causative affix, and the resulting values are assigned to the verbal stem. They percolate upwards and determine the shape of the tree. Thus, in line with the suggested analysis, the core has three direct arguments (consider feature-value pairs `arg: dir`). Also, in line with the suggested analysis, only one of them – the Actor – is assigned a macrorole. The other two arguments are non-macrorole bearing (consider `mr: nmr`).

In light of the research question of the current chapter, Zulu causatives can be fully related to Zulu (lexical) ditransitives since all of them feature a double-object symmetrical pattern. Thus, Zulu causatives belong to the type **Causative like ditransitive**. Technically, it is impossible to tell whether it is the subtype **Causee like Recipient** or **Causee like Theme** because all four participants – Recipient, Theme, Affectee, Causee – are marked identically. This is schematically shown in (78):

(78) $E_i^{caus}\ C_j^{caus}\ A_j^{caus}\ :\ A_i^{ditr}\ R_j^{ditr}\ T_j^{ditr}$

One might regard it as a syncretic strategy, not differentiating between the subtypes **Causee like Recipient** and **Causee like Theme**. Another approach could be to classify Zulu causatives as of type **Causee like Recipient** using statistical reasoning, since this subtype is more common than the other.

Another question is whether Zulu causatives have to be related to ditransitive constructions or to (mono)transitive ones. In fact, Zulu causative constructions comprise only two different ways of argument marking and thus can be related to bivalent argument marking patterns. Nevertheless, I am against this analysis. My first reason for that lies in the lexical dimension. Non-derived lexical ditransitives do exist in Zulu; hence, the way of argument marking they take can be considered somewhat "lexicalized". Thus, by relating causative derivations to ditransitives, one relates a derived construction to a lexical one, which does not seem to violate any methodological constraint. Second, the syntax of trivalent argument patterns differs from that of bivalent ones. Although both non-agent arguments in a trivalent pattern are marked identically to the non-agent of the bivalent construction, the very effect of symmetry emerges only once there is more than one object-like participant. Therefore, there is a difference between how objects in transitive constructions are marked and how participants of trivalent constructions behave. Given that, it is not sufficient to relate three-argument causative constructions to transitive ones in Zulu, but their relatability to ditransitives is legitimate.

6.2.1.2.5 Lubukusu: Two patterns Lubukusu is more complicated in this respect, as it has both symmetrical and asymmetrical causative constructions (consider (70)-(71) vs. (72)-(73)). The idea that Bantu languages need not be consistent in the symmetry of the double-object patterns they exhibit is discussed in Jerro (2015) and Van der Wal (2018). In a broader sense, regarding the coding of privileged syntactic arguments, this idea has been present in RRG from its early stages of development, bringing together various theoretical frameworks and allowing one to draw cross-theoretical conclusions. With respect to Lubukusu, the phenomenon of varied (a)symmetry is analyzed in Jerro (2019), where the author disagrees with Baker, Safir, and Sikuku (2012) and presents newly collected data from Kenya.

The present section aims to analyze both Lubukusu causative constructions, uncover the mechanism underlying their competition, and explain the factors that determine the choice of a given construction in a given situation.

According to Jerro (2019), two Lubukusu causative constructions are in complementary distribution: symmetrical patterns like (70) occur with ingestive verbs, while asymmetrical patterns like (72) are attested with predicates of other semantic classes (groups). The first step in studying both types of constructions would be to relate them to a ditransitive construction (and perhaps to discover some similarities with fully symmetrical languages like Zulu).

According to both papers (Baker, Safir, and Sikuku 2012, 57, fn.5 and Jerro 2019, 228, fn.13), there is one single ditransitive verb in Lubukusu. It demonstrates a symmetrical argument structure, i.e., a double object construction, from which either the recipient or the theme can be extracted as the new PSA. As shown in Sikuku, Diercks, and Marlo (2018, 406), this verb accepts either linear order of the objects. Also, either of the objects can appear on the verb as a bound morpheme. Thus, one can definitely say that the single non-derived ditransitive construction in Lubukusu has a symmetrical double-object argument structure. Subsequently, causative constructions with symmetrical patterns can be classified as type **Causative Like Ditransitive** (like in Zulu).

To explain why ditransitive constructions and causatives of ingestive verbs look alike, Jerro (2019) suggests a semantically-based account. Although it is not the main focus of the current section, I summarize it below, as it provides some insights into the (macro)role structure of Lubukusu symmetrical causatives.

In Jerro (2019), the author claims that Lubukusu ingestive transitive verbs are inherently ditransitive and thus involve three participants: the food, someone who manipulates it, and someone who (potentially) digests it. Moreover, they are claimed to be inherently causative: the event of manipulating the food causes the event of digesting it. With verbs like 'eat' or 'drink', the agent of the causing event (of manipulating) is identical to the agent of the cased event (of digesting). The patient of both events is also identical, it is the food. This, according to Jerro (2019), leads to the conclusion that these verbs have two syntactic arguments.

The causativization of ingestive verbs, according to Jerro (2019), does not introduce a new participant (or subevent) into the semantics of the verb. Instead, it allows the agents of the two subevents (manipulating food and digesting) to differ. The absence of this constraint opens a new syntactic valency, making it possible to express both the causer and the causee. The patients of both subevents remain identical – it is still the food that is manipulated and digested. Hence, the affectee is the third syntactic argument of the construction with a causative-of-ingestive.

The two syntactic non-subject participants in a causative-of-ingestive have equal object-like properties, hence the symmetrical behavior of such constructions demonstrated in (70)-(71). Applying the analysis of symmetrical constructions demonstrated in Section 6.2.1.2.4, one should admit that syntactically transitive ingestive verbs (like 'eat') and their three-argument causative counterparts (like 'feed') are M-intransitive. The same seems to be true for the lexically ditransitive Lubukusu verb 'give'. Thus, Lubukusu symmetrical patterns can be analyzed similarly to Zulu patterns. The tree for (70) would thus look like the Zulu tree shown in Fig. 12.

If the ditransitive analysis by Jerro (2019) is adopted, the status of constructions with caused ingestive verbs like (70) is somehow borderline between derived and non-derived. The range of the participants involved remains unexplored, but at the syntactic level, a new argument appears. Also, the verbal morphology indicates that a derivation has taken place. From the semantic side, Lubukusu symmetrical causative verbs are not related to ditransitive ones; they, in fact, **are** ditransitive. Consider the claim in Jerro (2019, 227): "Because caused ingestive verbs have a non-derived ditransitive event structure, it is expected that other semantically ditransitive verbs should pattern similarly with respect to symmetry."

This latter claim suggests that asymmetrical causatives in Lubukusu are not "semantically ditransitive" in Jerro's terms. Since the verbs appearing in asymmetrical causative constructions definitely outnumber those taking symmetrical patterns (according to Jerro (2019), these are all non-ingestive verbs), there must be a powerful mechanism in the Lubukusu language to produce them. Moreover, since symmetrical causatives appear only with a closed list of verbs and require very special semantic analysis, they can be regarded as somehow "marginal". In contrast, the asymmetrical causative pattern seems to be "default".

Since asymmetrical Lubukusu causatives cannot be related to ditransitives, perhaps they look like other common three-argument constructions? One might think of an instrumental construction, since instruments are often regarded as core arguments. In fact, there is no instrumental construction in Lubukusu that would have a non-derived verb. To license the thematic role of an instrument, an applicative suffix must modify the verbal stem. These applicative constructions are also asymmetrical, see (79) demonstrating the grammaticality of only one passive of the instrumental applicative.

(79) a. Lubukusu

Si-kombe	sj-a-fun-il-w-a	lu-sala	ne	omw-ana
c7-cup	c7sbj-pst-break-appl-pass-fv	c11-stick	by	c1-child

'The cup was broken with a stick by the child.'

b.

*Lu-sala	lw-a-fun-il-w-a	si-kombe	ne	omw-ana
c11-stick	c11sbj-pst-break-appl-pass-fv	c7-cup	by	c1-child

'The stick was used to break the cup by the child.' Jerro 2021, p. 30, (44)

The same applicative suffix can also introduce beneficiaries and locations. It is shown in Jerro (2015, 29–32) that instrumental and locative applicatives in Lubukusu are asymmetrical, while the benefactive applicative is symmetrical, see (80).

(80) a. Lubukusu

Omw-ana	k-∅-ombakh-il-w-a	en-ju	ne	omw-ami
c1-child	c1sbj-pst-built-appl-pass-fv	c9-house	by	c1-chief

'The child was built the house by the chief.'

b.

En-ju	y-∅-ombakh-il-w-a	omw-ana	ne	omw-ami
c9-house	c1sbj-pst-build-appl-pass-fv	c1-child	by	c1-chief

'The house was built for the child by the chief.' Jerro 2021, p. 29, (40)

Without diving too deep into the properties of the applicatives, I would like to point out that asymmetrical causatives can be related to instrumental and locative applicative constructions. The correspondence between argument coding is schematically shown in (81), where *j* stands for "object-like, has more object properties" and *k* stands for "object-like, has fewer object properties".

(81) $E_i^{caus}\ C_j^{caus}\ A_k^{caus}\ :\ A_i^{instr}\ P_j^{instr}\ I_k^{instr}$

Given the differences between the two object-like participants in asymmetrical causatives and instrumental (and locative) applicatives, one might suggest that they are not equally eligible for passivization, pronominalization, and other operations often used as tests for (a)symmetry. As the analysis of symmetrical constructions suggests, Bantu languages must be ignorant of the macrorole status of the object to allow its functioning as a "full object", i. e., to become PSA of the passive construction, pronoun, etc. Thus, the crucial difference between the causee and the theme in asymmetrical causatives like (72) lies not on the macrorole level. On the other hand, the applicative derivation of instrumental constructions implies that the instrument must be a core argument of the verb. The only difference left between the instrument as the "applied object" and the patient is thus the difference between direct vs. oblique core arguments. Thus, the definition of the full object should be specified as "any non-Actor direct core argument". The M-transitivity of the base verb does not matter because whether the causee bears a macrorole or not, it is predicted to exhibit the same syntactic behavior.

In the asymmetrical construction, one can observe a clean step-by-step demotion, i.e., each participant goes one position lower than its initial position. The hierarchy in syntactic terms looks like *"PSA → Object that can become PSA → Object that cannot become PSA"*. It is in line with Comrie's hierarchy suggested for the "paradigm case". What is different is that the former Subject does not "hop over" the DO position occupied by the former DO. Instead, all participants change their positions. In this respect, Lubukusu demonstrates an important difference from Finnish, where the causee (not the Theme!) receives the same coding as the prototypical instrument. To relate Lubukusu asymmetrical causative constructions to something less complex in terms of syntax, one could say that the causer and the causee show the same marking pattern as a prototypical agent-patient pair in a transitive construction. In contrast, the theme of a causative is demoted in the position that can be occupied by an additional argument in a non-benefective applicative construction.

6.2.1.2.6 Summary To summarize the analysis suggested in this section, I claim the following. In Zulu and Lubukusu, the causative derivation does not affect the M-transitivity. The difference between M-transitivity and S-transitivity greater than one is allowed. In Lubukusu, verbs of ingestion are M-intransitive. Once causativized, three-argument M-intransitive verbs have two postverbal object-like NPs, neither of which is assigned a macrorole. This results in their symmetrical behavior. In Zulu, all verbs behave like that. In contrast, non-ingestive verbs in Lubukusu do not allow three direct core arguments. When causativized, they exhibit a gradual demotion of all the participants: the causer occupies the PSA position, and bears the Actor macrorole, the former agent (i. e., the causee) becomes the direct core non-macrorole argument (able to take part in syntactic processes), finally, the former patient is demoted to the position of an oblique core argument (not eligible for syntactic processes). The same pattern is also attested for instrumental and locative applicatives in Lubukusu.

6.2.1.3 Challenge 2: Locus of marking

The previous section examined the challenging coexistence of two distinct double-object causative constructions. However, it is not the only challenge with Bantu three-argument causatives.

So far, only examples with full NPs and their properties have been examined. If one looks at pronominal arguments, one can observe that they are attached to the verb. Since Bresnan and Mchombo (1987), there has been a debate about whether pronominal arguments in Bantu should be analyzed as instances of head-marking. This section examines pronominal arguments in two Bantu languages and provides an RRG analysis.

6.2.1.3.1 Kinyarwanda: Data and background In many Bantu languages, verbs can have up to two morphemes related to pronouns. Although they look alike, their properties are quite different.

(82) Habimana y-a-men-esh-eje umw-ana igi-kombe — Kinyarwanda
Habimana c1.sbj-pst-break-caus-pfv c1-child c7-cup△
'Habimana made the child break the cup.' — Jerro 2017, p. 753, (4a)

(83) Umw-arimu y-a-ki-men-esh-eje umw-ana — Kinyarwanda
c1-teacher c1.sbj-pst-c7.obj-break-caus-pfv c1-child△
'The teacher made the child break it (i.e. the cup).' — Jerro 2017, p. 761, (20b)

The subject morpheme (*y-* glossed as *c1.sbj* in (83)) is obligatory. The subject noun phrase (*umw-arimu* 'teacher') can be omitted without making the sentence ungrammatical if and only if the referent can be recovered from the context. This is common for many Bantu languages (Nurse and Philippson 2003, 122), Kinyarwanda being among them (Jerro 2017, 754).

In the classical work Bresnan and Mchombo (1987), sbj morphemes in Bantu languages are referred to as 'anaphoric agreement'. This term highlights that the participant to which the prefix refers must be available in the previous context (anaphoric) and that the marker must share some grammatical features with the noun (agreement). (See Kihara (2017, 57–58) for an extended terminological discussion.) On the one hand, the behavior of the subject prefix can be related to the definition of an argument in head-marking languages: its omission makes the sentence ungrammatical, but the lexical specification of this participant is not obligatory. In prototypical head-marking constructions, however, **any** NP can be omitted, disregarding its appearance in the context.

Thus, the Kinyarwanda subject morpheme is not the clearest example of an argument affix and might need to receive a different analysis.

On the other hand, the subject prefix in Bantu looks like the verbal agreement in predominantly dependent-marking languages, e.g., Latin or Russian (see (87) and (88) several pages below). The status of the verbal agreement is debated in various studies about the locus of marking (for instance, Haspelmath (2014), see details in Section 2.4.5). So far, in RRG, instances of verbal agreement have not been treated as instances of head-marking. Even though this analysis is not fully justified from a comparative perspective (as criticized in Haspelmath (2014)), it allows a simple, concise analysis of individual languages.

The behavior of the object prefix (*-ki-* '*C7.OBJ*' in (83)) varies across Bantu languages (see details in Nurse and Philippson (2003, 124–125)). In Kinyarwanda, as (82) shows, it does not appear on the verb if the respective full NP is present in the sentence. Consider the difference between (83) and (82): the former having a *C7.OBJ* verbal prefix, but no *C7* NP, and the latter having the full NP *C7-cup*, but no *C7* verbal prefix.

Thus, the verbal object prefix and the full object NP are in complementary distribution. Bresnan and Mchombo (1987) calls Kinyarwanda object markers 'incorporated pronouns'. In prototypical head-marking constructions, however, noun phrases appear in addition to argument affixes. Moreover, in classic RRG, these are the argument affixes that are considered arguments of the clause, not full NPs. Hence, according to RRG, argument affixes cannot be omitted. Thus, the head-marking analysis does not hold for Kinyarwanda sentences like (82) and (83).

Thus, sentences like (83) cannot be analyzed straightforwardly using the dependent-marking approach because they have a pronominal argument within the verb, but the head-marking approach is also not applicable to them. Therefore, some modifications of either line of analysis are needed to cover this type of sentence.

6.2.1.3.2 Kikuyu: Data and a prior RRG analysis

In RRG, the problem of argument-related verbal morphemes has received special attention in Kihara (2017). This dissertation presents aspects of Kikuyu morphosyntax (also known as Gikuyu).

(84) a. Mo-thinj-i a-ra-thenj-ir-ɛ mbori — KIKUYU
c1-slaughter-NMLZ c1-PST-slaughter-PFV-FV c9.goat△
'The slaughterer slaughtered the goat.'

b. A-ra-thenj-ir-ɛ mbori
c1-PST-slaughter-PFV-FV c9.goat△
'S/he slaughtered the goat.'

c. A-ra-me-thenj-ir-ɛ mbori
c1-PST-c9OBJ-slaughter-PFV-FV c9.goat△
'S/he slaughtered it.'

d. *Mo-thinj-i a-ra-me-thenj-ir-ɛ mbori
c1-slaughter-NMLZ c1-PST-c9OBJ-slaughter-PFV-FV c9.goat△
'The slaughterer slaughtered the goat.' Kihara 2017, p. 56, (3.2)

In Kikuyu, subject and object verbal prefixes demonstrate exactly the same behavior as in Kinyarwanda. Therefore, the approach used in Kihara (2017) can be applied to analyzing Kinyarwanda (as well as other Bantu languages with the same patterns).

After offering an extensive review of prior approaches (especially terminological) to subject and object verbal prefixes in Bantu (Kihara 2017, 56–58), the author comes to the following conclusions of their role in a Kikuyu transitive clause (Kihara 2017, 58):

(85)

1. the subject marker is "a pronominal anaphor (PA) in Gĩkũyũ, and the first direct core argument",

2. the object marker is

   (a) "a "pronoun" (Pro), the second direct core argument in the absence of a lexical object NP ";

   (b) "in the absence of the pronoun, the NP is the second direct core argument".

This analysis relies on the premise that Kikuyu is (at least, to an extent) a head-marking language. The first claim in (85) suggests that the subject prefix, not the subject NP, is the first direct core argument. This reflects what is commonly found in head-marking languages (or constructions). The second claim in (85) tries pursuing the head-marking analysis and treats an object prefix as a core argument as well. The digression from the head-marking line of analysis is allowed only if there is no object marking at all.

While examining Kikuyu ditransitive constructions (including applicatives), Kihara (2017, 60–61) continues with the head-marking analysis. The author treats the subject marker as the true core argument and the full NPs for the recipient and theme as core-internal arguments as well. In this perspective, the subject NP is treated as an Extra Core Slot (ECS), according to the classic RRG analysis of head-marking languages (Van Valin 2013). An example of a ditransitive construction is given in (86), and its tree representation suggested by Kihara (2017) is shown in Fig. 13 (I only added the glosses to the original Figure for the sake of readability).

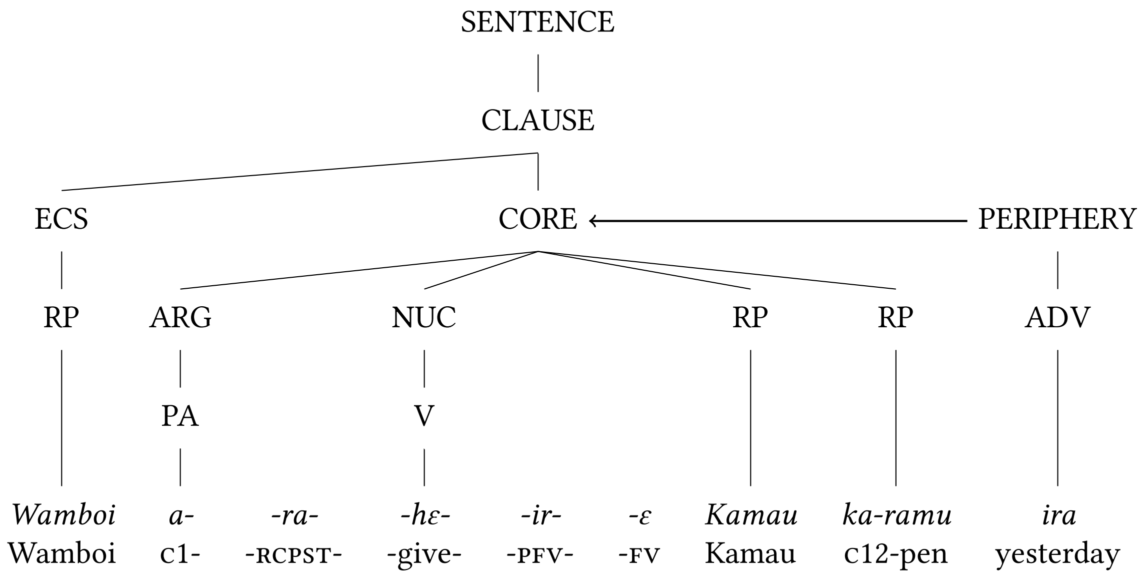


Fig. 13. The tree representation of the Kikuyu ditransitive (86) (reproduced from Kihara 2017, 60, Fig. 3.2)

(86) Wamboi a-ra-hɛ-ir-ɛ Kamau ka-ramu ira — Kikuyu
Wamboi c1-rcpst-give-pfv-fv Kamau c12-pen yesterday
'Wamboi gave Kamau a pen yesterday.' — Kihara 2017, p. 60, (3.5a)

According to the author, "*Wamboi* (together with the pronominal anaphor) is an agent" (Kihara 2017, 60), which is typical for head-marking constructions (even though the co-reference is not indicated in the tree). The structure of the CORE in Fig. 13 is, however, non-uniform. The pronominal anaphor (pa), being an argument of the verb, attaches to the CORE node, together with the NUC and the RPs (NPs). This is not typical for the classical RRG analysis of head-marking languages. The classical RRG analysis postulates the layered structure of the word and presupposes that there is a layer of the clause to which all the morphemes of a single word attach (consider Fig. 4). In the analysis suggested in Fig. 13, however, there is no such layer. Thus, the bound morpheme and the RPs are equal in their syntactic status, which speaks against the language reality.

6.2.1.3.3 A modified RRG analysis In my solution, I would like to modify the analysis suggested by Kihara (2017). The very idea of Van Valin (2013) was to account for sentences that have no NPs and to determine the status of the NPs once they appear in the sentence. However, mixing bound morphemes and independent NPs in order to obtain the desired number of arguments in the sentence looks like violating both lines of analysis: the traditional one (dependent-marking) and the head-marking one. Thus, I argue against the first claim in (85) and suggest analyzing Kikuyu (and Kinyarwanda) subject markers similarly to the verbal agreement in dependent-marking languages. I partly agree with the second claim in (85) and regard the object marker as an instance of incorporation. The section below presents this analysis in more detail.

In fact, all three NPs in Gikuyu and Kinyarwanda ditransitive constructions have different properties. They are summarized in Tab. 17. From this summary, it stands out that there is no clear argument of a head-marking construction, but at least one (Recipient) argument of a dependent-marking construction.

Since one of the arguments is never marked on the verb, I find it especially wrong to analyze these constructions from a head-marking perspective. Instead, I would suggest analyzing subjects in light of verbal agreement and objects in light of incorporation. The subject is the only argument in Kikuyu that necessarily has a class-corresponding morpheme on the verb. This can be related to verbal agreement in languages like Latin. Compare (87) below with (84a)-(84b).

(87) a. miles capra-m cecīd-it — LATIN
warrior goat-ACC slaughter.PRF-3SG
'The warrior slaughtered the goat.'

b. capra-m cecīd-it
goat-ACC slaughter.PRF-3SG
'S/he slaughtered the goat.'

c. ea-m cecīd-it
3SGF-ACC slaughter.PRF-3SG
'S/he slaughtered her (= the goat).' — Dr. Daria Kohler, personal communication

With this analysis, Latin sentences like (87a) and, respectively, Kikuyu sentences like (84a) are analyzed as having a subject realized as a full NP. All other NPs in a sentence are thus considered to be arguments as well, which keeps the analysis consistent. The verbal prefix *a-* 'c1' in (84b) can be thus related to the verbal ending *-it* '3SG' in (87b).

Sentences like (87b) are common in Latin as well as in other predominantly dependent-marking languages. In RRG projects that analyze data from different text types, the lack of an overt subject NP is not considered a complication for the analysis. Consider the syntactic representation of a sentence from the Russian translation of "1984" by G. Orwell, suggested in Bladier et al. (2022, (227ru)), shown in the original RRGParBank notation in Fig. 14. The sentence is accompanied by glosses and translation in (88).

	Subject	**Theme**	**Recipient**
place of the NP	pre-verbal	post-verbal	post-verbal
can NP be omitted?	no	must be replaced	yes
is cross-referenced on the verb?	yes	no	no
when NP is not there, does a verbal marker appear?	yes, it remains	yes	no

Table 17: Properties of arguments in Kikuyu ditransitive constructions

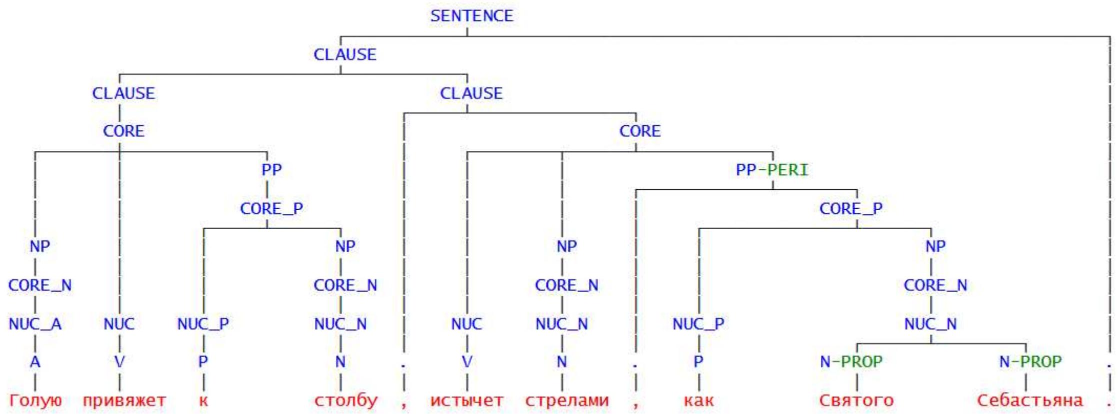


Fig. 14. A Russian sentence without an overt subject NP (taken from RRGParBank)

(88) Russian

Gol-uju	priv'až-et	k	stolb-u,	istyč-et	strel-ami,
naked.ADJ-ACC.SG.F	tie-3SG.FUT	to	stake-DAT.SG.M	pierce-3SG.FUT	arrow-INS.PL

kak	Sv'at-ogo	Sebastjan-a
like	Saint.ADJ-ACC.SG	Sebastian-ACC.SG.M

'He would tie her naked to a stake and shoot her full of arrows like Saint Sebastian.'[21]

In this sentence, there are two cosubordinated clauses. In the first clause, there is a transitive verb *priv'až-et* 'tie-3SG.FUT' and an accusative object *gol-uju* 'naked.ADJ-ACC.SG.F', but no subject. In the second clause, there is a transitive verb *istyč-et* 'pierce-3SG.FUT', the object of which is recoverable from the context (*gol-uju* 'naked.ADJ-ACC.SG.F'). The subject is still missing. The verbal ending *-et* tells that the subject is third person singular. In this particular case, it is discoverable from the broader context, namely, the preceding sentence.

The situation depicted in (88) and Fig. 14 is typical of pro-drop languages such as Latin and Russian. This trait aligns with Jerro (2017) 's account of Kinyarwanda, reinforcing the line of analysis I am suggesting. Hence, Kinyarwanda and Kikuyu subject marking can be analyzed like the verbal agreement in dependent-marking pro-drop languages.

What still requires analysis is Kikuyu sentences of type (84c) where the direct object is realized as a pronoun. In line with the analysis of the subject morpheme as an agreement affix, one could presume polypersonal agreement in Kikuyu and Kinyarwanda. However, prior research and the ungrammaticality of examples like (84d) argue against

[21] The treebank entry for the Russian sentence: https://rrgparbank.phil.hhu.de/editor/edit?sentno=227&lang=ru&context=browse#gold; the treebank entry for the aligned English sentence: https://rrgparbank.phil.hhu.de/editor/edit?sentno=228&lang=en&context=browse (accessed on 01.05.2026).

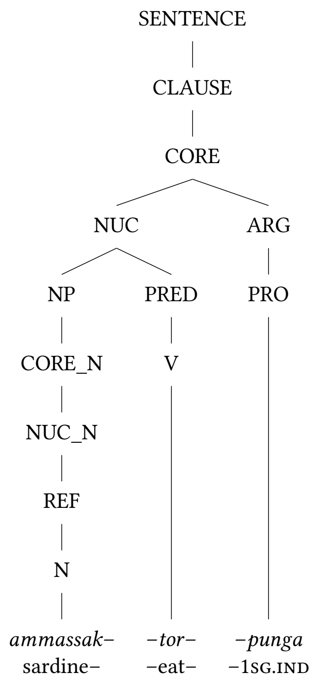


Fig. 15. A classic RRG analysis of noun incorporation in Greenlandic Eskimo

this idea.

If one compares Kikuyu (84c) to Latin (87c), one can see that the object marker *me-* is parallel to the free-standing pronoun *eam*. Given that, one could suggest that the object marker *me-* is a full pronoun but incorporated into the verb. This is in line with the classical analysis by Bresnan and Mchombo (1987), who call Bantu object markers 'incorporated pronouns'.

The classical RRG offers an analysis of noun incorporation, see Fig. 15 reproduced from Van Valin and LaPolla (1997, 65). Incorporated nouns in Greenlandic Eskimo keep their referential properties and thus are incorporated into the NUC as full NPs. Given the complementary distribution of object markers and full NPs in Kikuyu, I find this analysis specifically useful. One can thus formulate an alternation between pronominal (84c) and nominal (84b) objects in terms of tree government rules: if the object is a noun, it attaches below CORE, if it is a pronoun, it attaches below NUC.

6.2.1.3.4 Summary I suggest analyzing both Kikuyu and Kinyarwanda three-argument causative constructions as having free NPs as arguments, the subject requiring obligatory agreement on the verb. In this respect, they are relatable to **ditransitive** constructions. In case one argument is realized as a pronoun, two modifications follow. First, the structure becomes asymmetrical, and this argument gets realized as a verbal morpheme and attaches to the NUC node in the tree. Second, automatically, this participant receives the Undergoer macrorole.

In terms of tree representations, the alternation between a full noun in (82) and an incorporated pronoun in (83) can be modeled within the same family of trees, i.e., within the same line of analysis (see details in Part III). In contrast, using a tree from a head-marking family for (83) and a totally different tree for (82) seems to be unnecessarily complicated and not quite salient. This reinforces the decision to avoid analyzing Kinyarwanda and Kikuyu causatives as head-marking constructions.

6.2.2 Candidates for double subjects

6.2.2.1 General concerns

Dixon (2000) cites two languages where the A of the base transitive verbs retains its A-marking, and the newly added causer also receives A-marking. This situation is challenging to explain, since the theories usually predict that two equal agents cannot be in the same clause. Moreover, the definition of the causative construction I use is formulated in such a way that it requires the causer to be the highest argument of the clause.

> With respect to the structure of the caused event, causativization entails the introduction of a new subject, i. e., the causer of the matrix predication, into the underlying structure of the clause. This, in turn, forces an alteration in the status of the subject of the original clause. Kulikov (2001, 889)

This section presents the two languages cited as Type (ii) in Dixon (2000): **Trumai** (an isolate spoken in Brazil) and **Kabardian** (a language of the Abkhaz-Adyghe family spoken in the Kabardino-Balkarian Republic and partly in the Karachay-Cherkess Republic, Russian Federation). Examples (89)-(90) show causative constructions in these languages that conform to the description that both the causer and the causee have the "A-marking". Each of these examples has two participants who receive a case marker glossed as ERG.

(89) Alaweru-k hai-ts axos-∅ disi ka — TRUMAI
Alaweru-ERG 1-ERG child-ABS hit/kill CAUS
'Alaweru made me beat the child.' Guirardello (1999, p. 302, ftn. 2, (i))

(90) ł’əźə-m ś’āla-m pχa-r ∅-yə-ryə-ġa-q’^{w}ət-a-š’ — KABARDIAN
old.man-ERG boy-ERG tree-ABS 3SG.ABS-3SG.IO-3SG.A-CAUS-cut-PST-AFFMT$^{\triangle}$
'The old man made the boy cut the tree.' Matasović (2010b, p. 50, (18))

The analysis of these sentences is nevertheless not straightforward. First, one needs to ensure that the ERG marking is actually the "A-marking" in both these languages. Second, one needs to pay attention to the type of juncture in these constructions. One might hypothesize that the two A-like participants belong to different syntactic units and thus do not violate the causative definition.

Although genetically unrelated and geographically remote, Trumai and Kabardian demonstrate several similarities related to argument marking, especially in complex constructions.

Kabardian has a very limited set of markers available for marking core arguments; in fact, only two. Thus, a three-argument construction is already condemned to some morphological doubling in the absence of the diversity of devices eligible for such a function. In Trumai, the situation is more complex, but the choice is still quite limited.

Second, in both these languages, it is tricky to establish the basic alignment (see respective sections). This conundrum is reflected in the variety of glossing traditions for both these languages. The pair of cases shown in (89)-(90) can be referred to as "ergative-absolutive", "ergative-nominative", "oblique-absolutive", "oblique-nominative"... I stick to a certain tradition in each situation and explain my choice for each language in the respective section. What is nevertheless important is that the distribution of these core cases is not as straightforward as the standard definitions.

This challenge can be further developed: both Trumai and Kabardian exhibit unclear syntactic alignment. Recent studies in both languages (see references in the respective sections) question whether it is possible to claim at least partial syntactic ergativity for some constructions in these languages. Unfortunately, the scope of the present work prevents me from going deeper into this question, but the syntactic alignment remains very important for studying valency-related phenomena.

Moreover, both these languages have at least two basic bivalent constructions: one is traditionally referred to as "transitive", and the other is often called "extended intransitive". Importantly, causatives can be derived from both types of constructions in both languages. I examine how each pattern is built in each language in respective sections.

With all these challenging similarities in mind, Section 6.2.2.2 and Section 6.2.2.3 present the necessary data and observations drawn from descriptive studies, and Section 6.2.2.4 summarizes and discusses the conclusions.

6.2.2.2 Kabardian

6.2.2.2.1 Introduction Kabardian has a complex structure. In addition to case marking, Kabardian has head-marking, i.e., verbal prefixes indicating subject, direct object, and indirect object. As usual in head-marking languages, the verbal complex constitutes a grammatically correct sentence itself, the use of RPs is not obligatory.

In RRG, the verb in head-marking languages is treated as a CORE, with the verbal root (or stem) as the NUC and the bound morphemes as arguments. Free RPs are considered lexical specifications of the arguments. They are placed in Extra-Core Slots (ECS), special positions postulated for head-marking and double-marking languages that appear within the CLAUSE but not within the CORE. (See details and discussion of this approach in Section 2.4.5.)

Thus, the focus of the analysis of Kabardian causative constructions is twofold. First, one needs to examine the verbal structure to determine the properties of the true arguments, i.e., verbal prefixes. Second, one needs to explore the structure of the ECSs, where the RPs are placed, and explain the doubling of ERG case there.

Kabardian has already received attention in RRG studies. Ranko Matasović has published a grammar of Kabardian (Matasović 2010a) as well as some articles on Kabardian causative constructions (Matasović 2009; Matasović 2010b). However, to quote his own words, "[T]his grammar should be used with some caution, not only because it was written by a linguist who is far from being a fluent speaker of Kabardian. It is largely compilatory in nature..." (Matasović 2010a, 4). The main difficulty with these works is that the author pays almost no attention to head-marking, neither in the glossing of examples nor in the discussion of the suggested analyses. Therefore, in the present work, I also rely[22] on earlier descriptions of Kabardian, realized within different theoretical frameworks. In addition, in this section, I extensively quote the data and the descriptions of Adyghe, the closest relative of Kabardian (see Fig. 16). From the perspective of causative constructions, it is identical to Kabardian, but has received more attention in earlier studies and in recent typological scholarship.

6.2.2.2.2 The slot structure of the verb The verbs in head-marking languages are often described in terms of slots. All possible verbal affixes are listed in the linear order of their (possible) occurrence and are assigned a slot number. Naturally, some slots can be void, and some others cannot.

In Adyghe, there are six slots where arguments can be expressed. (I prefer the term "expressed" instead of "cross-referenced" since in RRG, verbal affixes, not RPs, are considered arguments.) Below, I follow the numbering by Smeets (1992): slots 1-6 counted from the beginning of the word. A different tradition counts affixes from the verbal root; hence, prefixes receive negative numbers. In this tradition (namely, in Arkadiev, Lander, et al. (2009, 42)), argument affixes are numbered -9 to -4.

For the study of three-place constructions, three argument slots are relevant. Slot 1 (or Slot -9) is obligatory for the PSA. It demonstrates ergative alignment, i.e., cross-references the S of an intransitive verb and the P of a transitive. If there is a corresponding RP in the sentence, it will receive ABS (*-r*) marking. This slot is obligatory for every verb; however, it is realized as a morphological zero '∅' in the third person (singular or plural).

The A of a transitive verb is realized as a Slot 6 (or Slot -4) affix. Of all argument affixes, the one denoting the most agentive participant is always closest to the verb. I

[22]I express my gratitude to Peter Arkadiev for drawing my attention to possible inaccuracies in Matasović's works and for offering me literature on several Abkhaz-Adyghe languages.

[23]Other sources, including Arkadiev, Lander, et al. (2009), offer a different list of Adyghe dialects. For the purposes of the current work, dialectal differences are barely relevant.

- Abkhaz-Adyghe family
  - Abkhaz-Abaza aka Abkhaz-Abazin
    - Abaza `abq`
      - Ashkaraua `ashk1247`
      - Bezshagh `besz1238`
      - Tapanta `tapa1256`
    - Abkhaz `abk`
      - Abzhui `abzh1238`
      - Bzyb `bzyb1238`
      - Samurzakan `samu1242`
  - Circassian
    - Adyghe aka West Circassian `ady`
      - Abadzex `abad1240`
      - Bezhedukh `bezh1247`
      - Natuzaj `natu1243`
      - Shapsug `shap1240`
      - Xakuchi `xaku1238`
    - Kabardian aka East Circassian `kbd`
  - Ubykh aka Ubyx `uby`

Fig. 16. The genealogical structure of the Abkhaz-Adyghe family[23](Hammarström et al. 2022; Eberhard, Simons, and Fennig 2021)

follow the glossing tradition used in Arkadiev, Lander, et al. (2009) (as well as other works by the same authors) and label this prefix as A, as shown in the examples below. This affix can be complemented by an ergative (-*m*) RP. Thus, prototypical transitive verbs in Adyghe have two argument prefixes, Slot 1 (-9) and Slot 6 (-4) (displayed as Pattern 1 in Tab. 18).

Slot 5 (-5) is used to encode an indirect object. The range of semantic roles eligible for this position varies, but the scope of this suffix broadly covers what is commonly referred to as an indirect object. Hence, the label IO in Arkadiev, Lander, et al. (2009) and similar works.

Slots 2 (-8), 3 (-7), and 4 (-6) are not relevant for studying causative constructions. It must be noted, though, that there are multiple alignments of bivalent and trivalent verbs, including different subsets of argument affixes. All these types are listed in Tab. 18.

An example of a sentence with a trivalent verb is given in (91), indices provided by the authors to explain the cross-reference. In (91), all RPs are unmarked, because "in Adyghe in general personal pronouns, proper names, and nouns marked with possessive prefixes do not take case-markers, and neither do case-markers attach to non-specific

NPs" (Arkadiev and Letuchiy 2009, 4).

(91) azamate$_i$ fatime$_j$ x'Ede$_k$ $\varnothing_k$-r$_j$-jə$_i$-ša-ʁ ADYGHE
Azamat(ERG) Fatima(ERG) donkey(ABS) 3SG.ABS-3SG.IO-3SG.A-sell-PST
'Azamat sold a donkey to Fatima.' Arkadiev and Letuchiy 2009, p. 3, (4)

	1	2	3	4	5	6	slot numbers in Smeets (1992)
	-9	-8	-7	-6	-5	-4	slot numbers in Arkadiev, Lander, et al. (2009)
1.	S					E	
2.	S				IO		
3.	S				IO	E	
4.	S			PO			
5.	S			PO		E	

Table 18: Basic bivalent and trivalent verbal alignments in Adyghe (from Smeets 1992, 127; modified)

In Kabardian, the system of the S (ABS), IO and A suffixes is basically the same (note quite a peculiar terminology and explanation offered in Colarusso (1992)). An example of a three-place verb is given in (92) (note that the ABS participant is not third person, hence, the suffix is not realized as a morphological zero). A three-argument construction with three RPs is shown in (93).

(92) wə-je-s-t-a-š' KABARDIAN
2SG.ABS-3SG.IO-1SG.A-give-PST-AFFMT△
'I gave you to him.'□ Kumaxov 2006, p. 165, (4)

(93) ataləqə-m səšx$_w$ə-r qanə-m ∅-yə[25]-ryə-t-a-sə' KABARDIAN
foster.father-ERG sabre-ABS adopted.son-ERG 3.ABS-3SG.IO-3SG.A-give-PST-AFFMT△
'The foster father gave (as gift) the sabre to his adopted son.'□
Kumaxov 2006, p. 435

6.2.2.2.3 Case marking Both Adyghe and Kabardian have a variety of morphological cases, but only two of them mark core arguments, see below.

> Specified (definite or indefinite) nouns in Kabardian can take four cases: absolutive, oblique, instrumental, and predicative. Two of these, however, the instrumental and the predicative, might be considered to be other than cases. The instrumental is actually a suffix on the oblique. At the same time, the predicative has a multitude of roles. However, it can generally be seen as an adverb suffix or a complementizer, indicating that the noun derives

[25]The cyrillic letter *u* is reproduced as *yə* following the transcription in Matasović 2010a. According to Kumaxov 2006, 54–55, the letter *u* stands for a variant of either a diphthong, or a short *e/ə*.

from an underlying absolutive or oblique that has been lost due to syntactic complexities. Colarusso (1992, 51)

The cases used for marking arguments have been glossed ABS (*-r*) and ERG (*-m*) so far, according to the most recent studies (such as Arkadiev, Lander, et al. (2009) and Arkadiev and Letuchiy (2009) *inter alia*). Other traditions use the terms Absolutive-Oblique (Colarusso 1992) or Nominative-Ergative (Kumaxov 2006; Matasović 2010a).

All studies attest systematic ergativity in Circassian languages in the system of verbal affixes and nominal case assignment. Nevertheless, syntactic ergativity remains a debated question, and its manifestation in valency-changing constructions receives special attention in Letuchiy (2012).

Thus, the ABS marker (*-r*) is used for S of intransitive verbs, P of transitive verbs, and Theme of ditransitives. In all cases, it denotes the less agentive and the most patient-like participant. In RRG terms, ABS verbal affixes would always receive the Undergoer macrorole.

The usage of the ERG (*-m*) nominal case is broader. Not only does it encode the most agentive participant in transitive and ditransitive constructions, but also the recipient of ditransitive verbs. Consider the information about Adyghe: "the Ergative functions not only as a marker of transitive subjects, but also marks all types of indirect objects, such as recipients of ditransitive verbs, or indirect objects of intransitive two-place verbs such as 'look' or 'read', as well as NP-internal possessors" (Arkadiev and Letuchiy 2009, 4). The following functions are reported for Ergative in Kabardian (quotes from Matasović (2010a, 19–20), which in turn relies mainly on Colarusso (1992, 53–54)):

1. "The ergative is, basically, the general oblique case used for all other [than S$_{intr}$ and P$_{tr}$] grammatical functions; it is the case of the transitive subject";

2. "The ergative can also correspond to the dative case in the European languages; it marks the recipient of the verbs of giving, and other oblique arguments";

3. "The ergative is also the case which marks the goal of the verbs of movement (like the Latin accusative of the goal)";

4. "The ergative can correspond to the locative case in those European languages which have it, indicating a spatial or temporal location";

5. "In some constructions, the ergative can correspond to the English possessive genitive or prepositional phrase".

Finally, Matasović (2010a, 20) concludes that the "ergative functions as both the case of the Agent and as a "general oblique case" covering all other functions of oblique and non-macrorole core arguments, but non-arguments (adjuncts) can also be in the ergative". One can easily note that these functions align with those listed for Adyghe.

base alignment	ABS_i — V_1	ABS_i — IO_j — V_2	ABS_i — PO_j — V_3
causative alignment	ABS_i — A_{causer} — V_1	ABS_i — IO_j — A_{causer} — V_2	ABS_i — PO_j — A_{causer} — V_3
base alignment	ABS_i — $\mathbf{A}_j$ — V_4	ABS_i — IO_j — $\mathbf{A}_k$ — V_5	ABS_i — PO_j — $\mathbf{A}_k$ — V_6
causative alignment	ABS_i — $\mathbf{IO}_j$ — A_{causer} — V_4	ABS_i — IO_j — $\mathbf{IO}_k$ — A_{causer} — V_5	ABS_i — PO_j — $\mathbf{IO}_k$ — A_{causer} — V_6

Table 19: Causative derivation in West Circassian (Smeets (1992, 127), modified)

The crucial point is that multiple ERG RPs are totally normal in Circassian languages, as both Agents and Non-macrorole core arguments are marked with this case by default. The case doubling does not mean doubling the same function or role, though, since two RPs are cross-referenced on the verb by different argument prefixes.

6.2.2.2.4 Causative verbs It is now time to explore causative constructions in Circassian languages. The causative prefix can be added to (practically all) monovalent, bivalent, and trivalent verbs. If the alignment pattern of the base verb already contains an A participant (realized as Slot 6), this participant gets demoted to the IO slot (5). In contrast, the causer gets realized as the new A argument. If the alignment pattern of the base verb has no A participant, it is just added. In Tab. 19, the correspondence between underived and causative verb structures is shown. The labels follow the glossing tradition used in (91), PO stands for "preverb object", Slot 4 in Smeets (1992). As the top rows of the table show, base verbs without an A participant (V_1, V_2, V_3) put the causer into Slot 6 without the demotion of the existent arguments. In the bottom rows with V_4, V_5, V_6, the former A participant is demoted from Slot 6 to Slot 5, while the causer takes Slot 6.

In this respect, the most interesting pattern is attested for V_5: it seems to have two indirect objects. However, this thesis will not study this pattern in more detail, as the scope of the present work includes only three-participant patterns.

All bivalent patterns, including the one commonly called *transitive* (V_4), demonstrate a clear instance of Comrie's paradigm case. When the highest position is available, it gets occupied; if already occupied, the demotion to the position of the indirect object, i.e., the highest free position, takes place. According to the classification offered in Chapter 6, Kabardian falls into the type **Causative like Ditransitive**, subtype **Causee like Recipient**.

6.2.2.2.5 Causative constructions: analysis and discussion Examples of Kabardian and Adyghe causative constructions derived from transitive verbs are shown in (94)-(95) respectively. For Kabardian, the initial transcription has been slightly modified, and the missing labels for the argument affixes have been added. Note the phonological identity of argument affixes in both these languages.

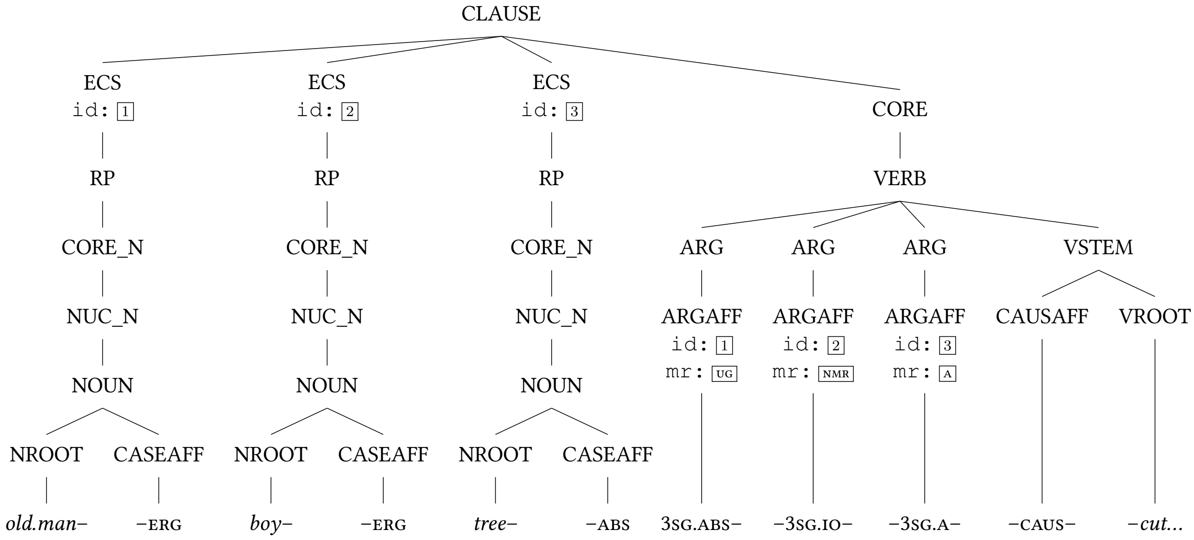


Fig. 17. The suggested syntactic analysis of Kabardian causative constructions

(94) ł'əźə-m ś'āla-m pχa-r ∅-yə-ryə-ġa-q'ʷət-a-š' Kabardian
old.man-erg boy-erg tree-abs 3sg.abs-3sg.io-3sg.a-caus-cut-pst-affmt△
'The old man made the boy cut the tree.' Matasović 2010b, p. 50, (18)

(95) jane pŝaŝe-m pisme ∅-r-jə-ʁe-txə-ʁ Adyghe
mother(erg) girl-erg letter(abs) 3sg.abs-3sg.io-3sg.a-caus-write-pst△
'Mother made the girl write a letter.' Arkadiev and Letuchiy 2009, p. 9, (22b)

In addition to three-argument causative verbs, both sentences have three RPs specifying the lexical meaning of the arguments. In Kabardian, all have overt case marking, while some zero morphological case marking is attested in the Adyghe example.

Prior RRG analyses of Kabardian (Matasović 2009; Matasović 2010b) treat such constructions as nuclear junctures and analyze them from the perspective of the *domain of case assignment* and *dependent-first case assignment pattern.* In a nutshell, the author claims that three-place causatives are core junctures, i.e., some instances of complex structures. The first CORE is the base verb, and the second CORE is produced by the causative morpheme. Thus, one of the erg RPs belongs to the upper CORE and the other – to the lower CORE. More details on this line of analysis are given in Section 6.2.2.3.4.1.

Even though it was developed for Kabardian, this analysis appears overly complicated when the data are analyzed with the necessary attention to head-marking properties. Kabardian causatives can be analyzed as simple clauses since all the participants are realized only once as verbal morphemes. The doubling of ergative RPs is due to the syncretic nature of the erg marker, which can mark a variety of participants.

The (simplified) syntactic structure of (94) is shown in Fig. 17 (operators are not annotated for the sake of better readability). The verbal root, together with the causative morpheme, constitute a single nucleus, the same way as in all dependent-marking languages explored so far. In contrast, the arguments of the CORE are below the VERB level because they are bound verbal morphemes. Since the verbal complex is a sufficient and grammatical sentence, these affixes are true arguments and thus are assigned macroroles. Full RPs are optional and are thus placed in the Extra-Core Slots (ECS). Each noun with an overt case marker receives the value of the case feature from the case morpheme. There is also the feature `id` assigned to both RPs and argument affixes. Thanks to this feature, one can ensure correct cross-reference between the RPs and the verb. More details on the technical realization of this solution are given in Section 10.4.2.4.

6.2.2.3 Trumai

Trumai is an indigenous language spoken by 51 (Guirardello 1999, xvi) people living in the Xingu Reserve in Brazil (more ethnographic information to be found in Guirardello (1999, xiv–xvii)). It is considered to be a genetic isolate. The main (early) stages of Trumai studies are listed in Guirardello (1999, xvii–xviii) and the latest approaches are named in Monod Becquelin and Becquey (2012, 290, fn. 1-2).

Apparently, the conclusion about Trumai causatives having two participants with "A-marking" has been drawn from example pairs like (96) vs. (97) – (98).

(96) ine-k atlat-∅ mapa — Trumai
3-ERG pan-ABS break
'He broke the pan.' — Guirardello 1999, p. 257, (4)

(97) hai-ts chï_in Atawaka-k atlat-∅ mapa ka — Trumai
1-ERG FOC/TNS Atawaka-ERG clay.pan-ABS break CAUS
'I made Atawaka break the clay pan.' — Guirardello 1999, p. 303, (3a)

(98) Alaweru-k hai-ts axos-∅ disi ka — Trumai
Alaweru-ERG 1-ERG child-ABS hit/kill CAUS
'Alaweru made me beat the child.' — Guirardello 1999, p. 302, ftn. 2, (i)

Example (96) shows a construction with a prototypical transitive verb, where the semantic agent receives the case traditionally glossed as ERG, and the patient receives the morphologically unmarked case, traditionally glossed as ABS[26]. Example (97) displays a causative of the same base verb, where the causer receives the same ERG marking as the agent in (96) (note: *-ts* is an allomorph of *-k* used with first person pronouns). The causee retains the same ERG (*-k*) marking. The affectee receives ABS marking. Example (98) is to demonstrate that the FOC/TNS particle is not obligatory in a causative construction and does not influence the choice of case markers (Guirardello 1999, 302, fn. 2). Thus,

[26]Monod Becquelin and Becquey (2012, 291, (2)) offer an alternative glossing claimed to be shared by some researchers: the unmarked case is labeled NOM and the *-k* marker is assigned an OBL label.

causatives like (97) – (98) end up having two NPs with ERG marking, which is perceived as double A-marking by Dixon (2000, 48)

Prior analyses attest that the morphological causative construction (with or without the FOC/TNS marker) "is uncommon in Trumai" (Monod Becquelin and Becquey 2012, 308). Since two identically marked constituents can lead to a reference conflict, "speakers try to avoid the presence of two Ergative NPs in the construction if the information can be inferred from the context, or it does not matter who the causee is" (Guirardello 1999, 304). Omission of the causee is not uncommon in many languages and must definitely be taken into account when analyzing the construction. However, it is still a question why, if all three participants in the causative construction are overtly expressed, they receive this particular marking.

It is not possible to relate such constructions to those with instruments (99). Trumai has a specific instrumental case, which introduces adjuncts but can never appear on arguments. Morphologically, it is quite different from other cases; thus, no ambiguity or case syncretism is to be suspected.

(99) intsatske yaw cho xu-letsi de — TRUMAI
next people perforate needle-INS already
'And then one perforates with a needle.'
Monod Becquelin and Becquey 2012, p. 301, (33)

Passives do not exist in Trumai as a specifically marked verbal category, so the marking of the demoted agents in this situation is also out of consideration. Two ditransitive patterns in Trumai will be presented in Section 6.2.2.3.3, but there is no one-to-one correspondence between them and causative constructions.

A short note on the etymology of the Trumai causative marker is also important to make. According to Guirardello (1999, 312), the causative particle *ka* could have grammaticalized from the verb *kapan* 'make' and the causative of a transitive was initially bicalusal. However appealing this hypothesis might be, the same source (Guirardello 1999, 314) disproves it: "However, the construction **cannot be considered bicalusal anymore** (author's emphasis – VG), because of the current behavior of *ka*, which can no longer [be] classified as a verb". Repeating the same etymological argument, Guirardello-Damian (2003, 17) concludes that "the causative construction in Trumai would be an intermediate type between the analytical and the morphological causative". Thus, one can suspect some traits of a biclausal pattern in modern Trumai causative constructions, but should not consider them prototypically biclausal anymore.

Existing studies admit that "[i]n no other construction in Trumai, a single verb can have two A arguments" (Guirardello 1999, 311), where the term "A argument" stands for a phrase bearing the marker glossed ERG (*-k*). Therefore, it seems impossible to relate this construction to any other one. This situation is disturbing for a theory of argument-

Morpheme	Label	Usage (according to Monod Becquelin and Becquey (2012))
-∅	ABS	Encodes a proto-agent in a ‘‘pseudo’’ extended intransitive construction, or a proto-patient in the ergative construction and any role in the intransitive construction
-(V)k	ERG	Encodes not only the agent of ‘‘ergative’’ verbs but also the causer in causative constructions involving verbs of any class
-(V)tl	DAT	Individualized, identifiable and salient dative (after Guirardello-Damian (2004))
-ki	DAT	Individualized, non-identifiable and non-salient dative; non-individualized, identifiable and non-salient dative (after Guirardello-Damian (2004))
-(V)s	DAT	Non-individualized, non-identifiable and non-salient dative (after Guirardello-Damian (2004))

Table 20: Cases used for encoding core arguments in Trumai

marking patterns in causative constructions. In this section, I am going to analyze sentences like (97)-(98) in relation to different argument-marking patterns found in Trumai. Therefore, one needs to study those patterns in more detail first.

6.2.2.3.1 Case marking in Trumai Trumai is a language able to mark a variety of cases on an NP. Guirardello (1999, 74–75) and Monod Becquelin and Becquey (2012, 301–302) provide slightly different lists of cases, which might also be considered as postpositions. In the current work, it is crucial that three cases are used in basic bivalent and trivalent patterns, i.e., cases suspected of encoding core arguments (in RRG terms). These cases are listed in Tab. 20, along with additional comments on their use. The glossing of these cases is arguable (extensive discussion on that is offered in Monod Becquelin and Becquey (2012)). In this work, I decided to stick to the glossing traditions I find in most of my examples and use the labels provided in Tab. 20. Not only does this decision avoid additional modifications to the examples, but it also keeps the glosses easy to read (in contrast to leaving unlabeled morphemes that can be glossed differently).

Of all these cases (ABS, ERG, DAT) able to encode core arguments, ABS is opposed to ERG and DAT. According to Monod Becquelin and Becquey (2012), from the point of view of the coding, ABS seems to be the “syntactic subject” or at least demonstrate the greatest number of “subject properties”: “The unmarked argument is obligatory in every construction; it governs verbal inflection (*-n/-e* in the 3rd person); it is not marked morphologically; it represents the prototypical agent in the ‘‘extended intransitive’’ construction; it is the addressee of the imperative in the ‘‘extended intransitive’’ construction; it commands reflexive, reciprocal and middle use <...>; and it is placed according to syntactic constraints of linear precedence” (Monod Becquelin and Becquey 2012, 308–309). From the syntactic point of view, the participant receiving the ABS marking seems to be the most important since it cannot be omitted, cf. Monod Becquelin and Becquey (2012, 308): "all arguments apart from the unmarked one are only optionally expressed". Moreover, Monod Becquelin and Becquey (2012, 310–312) offers an analysis of topic chaining in a Trumai text and concludes that "[t]his study brings to light a strong correlation between the discourse topic and the unmarked argument" (Monod Becquelin and Becquey 2012, 310).

Evidence from the coding and, to some extent, evidence from the behavior of ABS suggest that it might be the PSA in a Trumai clause. If so, two ERG participants in causative constructions are definitely not two PSA-like participants. It sorts out the possible violation of the definition of a causative construction since the ERG marking is proved not to be the PSA marking. The term "A-marking" is ambiguous as it is unclear whether it refers to morphological encoding, syntactic properties, or semantic role. Thus, syntactic alignment in Trumai still requires a thorough analysis.

6.2.2.3.2 Two-argument patterns in Trumai Three basic cases yield two patterns of argument marking in bivalent constructions. The question of their transitivity remains open and has been debated extensively (mainly in Guirardello-Damian (2003) and Monod Becquelin and Becquey (2012)). For this section, the question of whether any of these patterns count as transitive is secondary to the fact that both of them are bivalent[27].

The first pattern has been shown in (96). It is often called "ergative" or "transitive" pattern. Typical of this pattern is the unmarked undergoer and the actor marked with ERG (*-k*). Verbs that have their arguments marked this way are grouped into *class 2* by Monod Becquelin and Becquey (2012) and *type 2* by Guirardello-Damian (2003). According to the lists provided in Monod Becquelin and Becquey (2012) and Guirardello-Damian (2003), as well as the Wordnet-driven analysis of Trumai lexicon in Monod Becquelin and Becquey (2012, 304–307), many verbs high in Transitivity (in terms of Hopper and Thompson (1980)) appear in this class alongside with "a large number of verbs for ''peaceful'' (i.e. not prototypically transitive) actions" (Monod Becquelin and Becquey 2012, 305).

The second pattern is demonstrated in (100)-(101). It is sometimes referred to as "extended intransitive", because the participant labeled as DAT is not considered to be an object.

(100) Kumaru-∅ sone wïrïx-es TRUMAI
Kumaru-ABS drink manioc.porridge-DAT
'Kumaru drank (some) manioc porridge.'□ Guirardello-Damian 2003, p. 15, (28a)

[27] It is worth mentioning that some verbs in Trumai can occur in both patterns. An explanation of the variable selection of a bivalent pattern from the pragmatic yet anthropological perspective is offered in Monod Becquelin and Becquey (2013).

(101) hine ma k'ate-s/-ki/-tl TRUMAI
3.ABS eat fish-DAT$_{1/2/3}$$^{\triangle}$
'He eats (the) fish.' Monod Becquelin and Becquey 2012, p. 291, (2c)

The unmarked case appears on the agent-like participant, while the patient-like participant receives DAT marking. Any of the DAT markers listed in Tab. 20 can appear in this position, as the disjunction in (101) shows. Neither of the studies (Guirardello-Damian 2003; Monod Becquelin and Becquey 2012) differentiates between them in this role. In most elicited examples, DAT NPs follow the verb, while ABS and ERG NPs precede it. However, corpora studies show that both DAT and ERG can appear in either preverbal or postverbal position or be omitted. Cf. Monod Becquelin and Becquey (2012, 307): "the absolutive argument <...> is <...> the only argument subject to a strict word order constraint". Thus, the word order criterion is not decisive for distinguishing between DAT and ERG participants.

Examples (100)-(101), in fact, are remarkable in a number of other aspects that prove to play a role in building causatives across languages. First, the base verb in each of them is a verb of ingestion. Verbs of this class have already been attested as demonstrating special behavior in causative construction in Bashkir (Section 6.1.1.4) and Lubukusu (Section 6.2.1.2.5). According to Monod Becquelin and Becquey (2012, Fig. 2, p. 306), ca. 90% of Trumai verbs of ingestion follow the basic bivalent pattern illustrated in (100)-(101) above. Second, it can be supposed from the translations and the known behavior of verbs of ingestion that the actions described in (100) and (101) can be construed as atelic (in RRG terms, activities, but not accomplishments) and that the participant marked with a DAT can be regarded as non-referential[28]. The source paper of (101) is written in English, and the authors provide the optional article '(the)' in their translation. It might tell that the participant *fish* can be regarded either as specific (*the fish*) or unspecific/non-referential (just *fish*). Either this contrast is irrelevant for these verbs, or the reading is ambiguous. As for (100), the source paper contains the translation in Portuguese "*A Kumaru tomou mingau*" without any article (although Portuguese has them). This is also in favor of the fact that the sentence at least *might* read as an activity with a non-referential patient. The non-referential participant often has fewer Undergoer-like properties and thus is not a fully formed object of a verb. Hence, its place in the semantic and syntactic structure differs from the place of a referential object. In this respect, the DAT marking of objects in (100)-(101) is cross-linguistically predictable.

Verbs having the ABS-DAT bivalent pattern are grouped into *class 3* by Monod Becquelin and Becquey (2012) and *type 4* by Guirardello-Damian (2003). Both studies observed that almost all verbs of cognition, perception, and consumption fall into this class. As Bashkir data show (Section 6.1.1.4), in causative constructions in some languages,

[28]Van Valin (2018b, 80) cites a Dyirbal example very similar to (100)-(101): an atelic reading of the verb 'eat' triggers the ABS-DAT argument marking.

	ERG-ABS pattern		ABS-DAT pattern	
	status of ERG	status of ABS	status of ABS	status of DAT
Ergative interpretation	Core Argument 1	Core Argument 2	Core Argument 1	Extended to core argument
Accusative interpretation	Oblique Argument	Core Argument 1	Core Argument 1	Core Argument 2

Table 21: Traditional interpretations of the Trumai verbal system

verbs of cognition and perception can demonstrate similar behavior as verbs of ingestion (consumption). However, this class also features many highly transitive verbs, such as *hunt*, *carry*, *pick up* (Monod Becquelin and Becquey 2012, 295, Guirardello-Damian 2003, 10).

To summarize Monod Becquelin and Becquey (2012), there are two traditional interpretations of the Trumai verbal system, see Tab. 21. Each of these interpretations treats only one of the bivalent patterns as "transitive", i.e., having two core arguments, and suggests some special approach to the other pattern. Apparently, neither of these interpretations is adequate to account for the complexity of the whole system.

It seems reasonable to treat both ERG and DAT participants similarly, opposing them to the ABS participant. One of the options (discussed in Monod Becquelin and Becquey (2012, 313–314)) would be to claim all Trumai verbs intransitive. The idea behind this interpretation is that "no overt distinction can be observed at the morphosyntactic level which could be used to differentiate between ergative/accusative/dative arguments and oblique arguments" (Monod Becquelin and Becquey 2012, 313). In fact, this difference can be found. The authors mention that only "agents and patients" (which translates to "ABS/ERG/DAT participants" in terms of this section) are accessible for valency reduction operations (see also Guirardello-Damian (2003, 19)). Thus, ERG and DAT are different from oblique cases and should somehow be included in the core of the verb. Finally, Monod Becquelin and Becquey (2012, 315) come up with the interpretation including two classes of transitive verbs (ERG-ABS and ABS-DAT), each having a "subject argument" (taking the unmarked case, ABS) and a "complement argument" (taking ERG or DAT respectively).

6.2.2.3.2.1 A possible RRG analysis of Trumai bivalent patterns The "all-intransitive" hypothesis suggested and rejected by Monod Becquelin and Becquey (2012) can, in fact, become a start for an RRG analysis of Trumai bivalent patterns. Since the ABS argument is so different from the others, one can suppose that it is the only macrorole-bearing argument. Naturally, it would take the Undergoer macrorole. If one assumes that Trumai tends to select the lowest macrorole as the PSA (and if one agrees that the ABS participant indeed is the PSA in all sentences), one would get a clear vision of the monovalent and bivalent alignment patterns. Intransitive static verbs (like 'sleep') have the sole Undergoer and thus mark it with ABS case. Intransitive dynamic verbs (like 'walk') have only the Actor, but since it is the only macrorole, it also receives ABS marking. "Transitive" verbs like 'break' in (96) follow the pattern: the lowest MR = the UG = the patient receives ABS, the other receives the Agentive case, ERG and, apparently, the Actor macrorole. In examples with "extended intransitive" verbs like (100), the patient (e.g., 'porridge') does not qualify as the lowest-ranking argument because it is non-referential. Hence, the other participants get assigned the macrorole and the ABS case, while the non-macrorole patient receives the DAT case. This analysis is, in fact, well compatible with the final interpretation offered in Monod Becquelin and Becquey (2012).

6.2.2.3.3 Three-argument non-causative patterns in Trumai Both Guirardello-Damian (2003) and Monod Becquelin and Becquey (2012) list three-argument non-causative verbs as subtypes of bivalent classes. Both sources agree on these verbs being trivalent, but the question of whether they are ditransitive remains open.

(102) Kumaru-k oke-∅ kiţï hai-tl — TRUMAI
Kumaru-ERG medicine-ABS give 1-DAT
'Kumaru gave me medicine.'□ — Guirardello-Damian 2003, p. 8, (13a)

Trivalent constructions like (102) have an ERG-marked agent, an ABS-marked theme, and a DAT-marked recipient. In respect of marking agents and patients, they look exactly like verbs in (96) (often called "transitive) and thus are sometimes referred to as "ditransitives".

In the meantime, constructions like (103) exist as well.

(103) ha deţne kain k'ate-s kiki wan-ki — TRUMAI
1ABS distribute FOC fish-DAT man PL-DAT△
'I am distributing fish to the men.' — Monod Becquelin and Becquey 2012, p. 297, (17)

One could call them "even further extended intransitives", because they repeat the ABS-DAT pattern of the so-called "extended intransitives" (100) and add yet another DAT participant to it.

It is important to note that all agent-like and patient-like participants retain the same case marking as in bivalent patterns, while recipients always receive DAT marking.

6.2.2.3.3.1 A still possible RRG analysis One can try to expand the analysis of bivalent patterns, which is based on the premise that the ABS participant is the PSA in Trumai, to explain trivalent patterns as well.

Sentences like (102) look like ditransitive constructions in most languages. The case and the macrorole assignment is done as follows: the lowest-ranking participant receives the Undergoer macrorole and the ABS case; the highest-ranking participant receives the Actor macrorole and the ERG case; the remaining participant is NMR (non-macrorole) and receives DAT as an 'elsewhere' case.

The situation with sentences like (103) is more difficult. The argument *ha* '1ABS' is the effector and thus is ranked very high on the Actor-Undergoer Hierarchy. Hence, it receives the Actor macrorole. However, it receives not the ERG case, as is suspected for Actors, but ABS. The only explanation for that is that this argument is both the highest and the lowest direct argument in this sentence. This can only happen if *ha* '1ABS' is the only macrorole-bearing argument in the sentence, i. e., that the verb *deţne* 'distribute' is M-intransitive in Trumai. The two dative NPs in (103) are thus both oblique core arguments (or maybe adjuncts, but definitely not direct core arguments). Given that, one could regard patterns like (103) as "even further extended intransitives".

6.2.2.3.4 Three-argument causatives in Trumai After examining Trumai bivalent and non-derived trivalent constructions, one can now turn to three-argument causative constructions. For the sake of convenience, references to the examples with both bivalent patterns (96) and (100) with their causative counterparts (98) and (104) are given in Tab. 22.

(104) hai-ts Kumaru-∅ sone ka wïrïx-es TRUMAI
1-ERG Kumaru-ABS drink CAUS manioc.porridge-DAT
'I made Kumaru drink (some) manioc porridge. ' Guirardello 1999, p. 302, (2a)

	Basic pattern	Causative pattern
"Transitive"	(96) ERG-ABS	(98) ERG-ERG-ABS
"Extended intransitive"	(100) ABS-DAT	(104) ERG-ABS-DAT

Table 22: Trumai bivalent patterns and their causative counterparts

Constructions like (104) have not been mentioned in Dixon (2000). However, they have an explicit causative morpheme and are built on bivalent bases, as shown above. Therefore, both constructions like (98) and (104) are of interest for the present study.

In both causative constructions, the newly added causer is bearing the ERG case. Whatever the initial pattern, the causer takes the clause-initial position. This trend is attested in the Trumai grammar: "[t]he causer always precede[s] the causee, and often is followed by the Focus/Tense particle" (Guirardello 1999, 302). As a result of the causer addition, constructions like (104) look like one type of underived trivalent constructions (102), and constructions like (98) look exceptional, having two ERG arguments.

One possible option would be to treat these two constructions differently. It is easy to claim that causative constructions of type (104) belong to the type **Causative like Ditransitive**. What is peculiar about this type is that the causee receives the same marking as the theme of the ditransitive construction, not the recipient. Consequently, it is the affectee of the causative construction that received the same marking as the recipient of the ditransitive. In that case, one would need to postulate in Trumai a presumably non-existing causative strategy **Affectee like Recipient**, see Tab. 23.

	Causer	Causee	Affectee
(104) (causative of ext. intr.)	1-ERG	Kumaru-ABS	porridge-DAT
(102) (ditransitive)	Kumaru-ERG	medicine-ABS	1-DAT
	Actor	Theme	Recipient

Table 23: Relation of type **AffecteeLikeRecip** between (104) and (102) in Trumai

Also disturbing is the resemblance of causatives with an "extended intransitive" base to non-causative trivalent verbs with the "transitive" agent-patient alignment. No clear pattern of derivation can be traced between these two constructions. Thus, one might suspect that these constructions' superficial similarity is due to very different underlying processes. If this analysis is adopted for constructions like (104), one still needs not only an interpretation of constructions like (98) but also a mechanism explaining the selection of either strategy.

What needs nevertheless to be observed – and is mentioned in Guirardello-Damian (2003, 14–15) and Monod Becquelin and Becquey (2012, 292) – is that both causative constructions are built in the same way with respect to the base pattern they are derived from. In both examples (98) and (104), all the arguments that were present in bivalent patterns ((96) and (100) respectively) have exactly the same marking. Therefore, the fact that causatives of kind (104) look like non-causative trivalent verbs with the "transitive" agent-patient alignment (102) is to an extent *accidental*. It is not the fact that the arguments in a causative construction derive in such a way to fit the trivalent pattern, but that the trivalent pattern looks like the ERG-ABS bivalent pattern with a DAT argument added.

An important claim follows from this observation: in neither Trumai causative construction does any argument demotion occur. So, the relationship between bivalent patterns and their causative counterparts can be formalized as shown in (105). To paraphrase (105) with words, there are two claims to be made, see (106).

(105) $A_i^{tr}\ P_j^{tr}\ :\ E_{erg}^{caus}\ C_i^{caus}\ A_j^{caus}$

(106) a. No matter how arguments in the bivalent pattern are marked, they retain their marking in a causative construction.

b. No matter how arguments in the bivalent pattern are marked, the causer always receives ERG (*-k*) marking.

This interpretation gets away from the interpretation of (98) and (104) as two different causative constructions in Trumai and offers evidence to a previously unattested type of argument marking in a causative strategy **Causative like Bivalent + given case**.

6.2.2.3.4.1 Background on the domain of case assignment The first idea that comes to mind once two RPs bearing the same case are found in a clause is to suspect some complex structure behind them. The two ERG RPs in examples like (97)-(98) would then belong to different units. The question is, what would those units look like? This question can be approached from the perspective of the *domain of case assignment* (Van Valin 2005, 257–260). This notion designates a layer of the sentence structure (core or clause) that is the scope of the application of case-marking rules. In simpler words, bringing RRG closer to other traditions, according to the classical RRG approach towards the domain of case assignment (Van Valin 2005, 257–260), there are two types of languages: in some languages, the commonly used notion of the "simple sentence" should be specified as "single clause sentence", in others – as "single core sentence".

The classical idea of the domain of case assignment has been criticized in Matasović (2009) and Matasović (2010b). According to this approach, all languages have CLAUSE as the domain of case assignment, but the order of application of case-marking rules differs across languages. Some languages first assign cases to the inner (also called *linked*, *embedded*) CORE and second to the outer (*linking*, *matrix*) one. Other languages, in contrast, first assign cases to the matrix CORE and afterward to the linked one.

It is important to highlight that both approaches are developed for CORE junctures. (See the details on all nexus-juncture types in Section 2.6) However, they do not specify what type of juncture qualifies as a multiple domain: coordination, subordination, or cosubordination. Also, it has never been stated whether any kind of similar effects can be observed for other rules and other layers of the syntactic structure.

6.2.2.3.4.2 An analysis of Trumai causatives inspired by the domain of case assignment The notion of the domain of case assignment could be very helpful for the analysis of Trumai causatives. However, before applying this analysis, one needs to make sure that Trumai constructions are core junctures. In Van Valin (2005, 188–189), CORE junctures are defined as follows: "In a core juncture, there are two nuclei, each with its own set of core arguments, constituting two distinct cores. In non-subordinate core junctures, they overlap in that the linked units obligatorily share one core argument. In contrast, in a subordinate core juncture, the linked unit serves as a core argument of the matrix verb."

For causative constructions, one would suspect a core subordination. This analysis is shown in Fig. 18. The causative verb (or morpheme) would be the matrix core, taking the causer as its first argument and the whole core of the base verb as the second argument. This hypothesis aligns well with the absence of demotion, as the arguments of the inner core are imported intact into the upper core. This analysis could then explain the duplication of ergative RPs. Since the domain of case assignment is CORE, only one ERG per CORE is allowed. The CAUS morpheme builds its own CORE and thus opens a slot for an additional ERG RP. The causer takes this slot, while the arguments of the embedded CORE are not demoted.

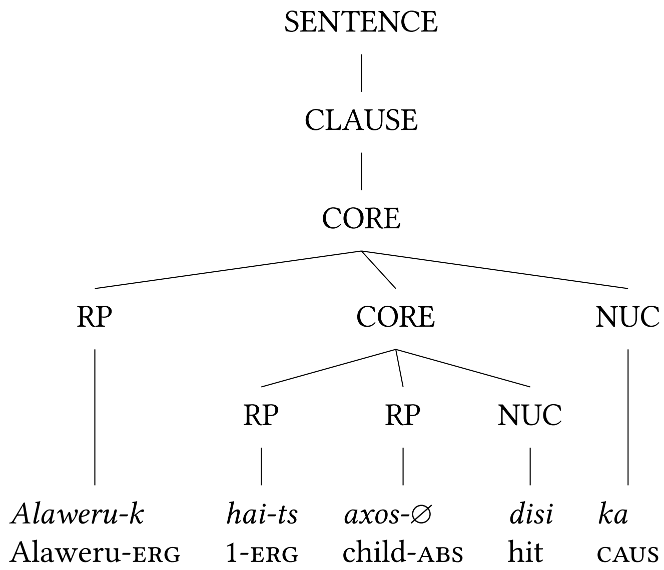


Fig. 18. The analysis of Trumai causative (98) 'Alaweru made me beat the child.' as core subordination

The same tree can be analyzed using the notion of the dependent-first case assignment pattern suggested in Matasović (2009). First, the transitive verb *disi* 'hit' builds its argument structure and uses an ERG-ABS pattern for that. Second, the transitive predicate *ka* 'CAUS' builds its argument structure and would use an ERG-ABS pattern as well if both arguments were RPs. However, the supposedly ABS argument is a whole core, in which all arguments retain their markers.

There is, however, one piece of descriptive evidence that casts doubt on the analysis above. According to Guirardello (1999, 307–308), the causative morpheme *ka* does not behave as a separate verb able to build its own CORE. Namely, it is impossible in the clause-initial position (107), whereas usual verbs (like *tao* 'order') can occur at the beginning of the clause (108).

(107) *ka ke ka_in hai-ts Atawaka-∅ huma — TRUMAI
CAUS KE FOC/TNS 1-ERG Atawaka-ABS take.bath△
*'I habitually/always make Atawaka take a bath.' — Guirardello 1999, p. 307, (18)

(108) tao ke ka_in hai-ts Atawaka-∅ huma — TRUMAI
order KE FOC/TNS 1-ERG Atawaka-ABS take.bath
'I always order Atawaka to take a bath.' — Guirardello 1999, p. 307, (14)

Moreover, the particle *(i)yi* (whose function is not totally clear, but is related to the notion of predication) can appear between the matrix verb and its complement, but cannot be inserted between a verb and the causative morpheme *ka*, see (109a) vs. (109b).

(109) a. *hai-ts ka_in Kumaru-∅ sa yi ka — Trumai
1-ERG FOC/TNS Kumaru-ABS dance YI CAUS
*'I made Kumaru dance'. — Guirardello 1999, p. 308, (20)

b. hai-ts chi_in axos-∅ waţkan yi homne
1-ERG FOC/TNS child-ABS cry YI find
'I found the child crying'. — Guirardello 1999, p. 308, (21)

In Guirardello (1999, 307), the author suggests that sentences with the verb *tao* 'order' can be considered analytical causatives, as they have a full verb with some causative meaning. They are, though, different from the constructions with the *ka* particle, which is not a full verb and has the grammatical meaning of introducing a cause exclusively. Given all these facts, the analysis of Trumai causatives shown in (97), (98), (104) as core subordination is compromised.

However strange the Trumai structure might seem, this fact is well predicted by RRG. According to the hierarchy formulated in Van Valin (2005, 206–207), the direct causation is the tightest type of all interclausal semantic relations. It thus must involve the tightest type of juncture of the lowest possible level. This type, as illustrated in Van Valin (2005, Fig. 6.22, p. 209), is nuclear cosubordination. As for core subordination, it is supposed to be the linking type in distant and permissive causatives. Even though special evidence stating that Trumai constructions in (97), (98), and (104) are factual or implicative is missing, they clearly are instances of a general type of construction, productive for all types of verbs and perhaps unmarked in terms of implicativeness (cf. Guirardello (1999, 304–307) discussion of analytical constructions used to express specified information together with the semantics of causation). Thus, the failure of the analysis of morphological Trumai causatives as core junctures is only logical and well-expected.

The last point to clarify is a possible idea to analyze sentences (97), (98), (104) as nuclear junctures and claim that Trumai would have an even narrower domain of case assignment. This analysis is totally impossible in RRG, while no argument assignment is done within the NUC by design.

6.2.2.3.4.3 Problematic RRG analysis as a three-argument clause Unfortunately, the previously suggested analysis inspired by the assumption of some kind of syntactic ergativity in Trumai (Section 6.2.2.3.2.1, Section 6.2.2.3.3.1) fails to explain causative constructions without a deep reanalysis of Trumai case marking and of the causative derivation itself. The problem with the simple-clause-like analysis of Trumai causatives is mainly the total lack of demotion. The addition of a participant to a simple clause without the demotion of the existing participants is possible when this added participant is either a non-macrorole core argument, or an oblique argument, or an argument-adjunct, or even an adjunct. None of these definitions can be used for the traditional causer in the traditional causative construction. To allow multiple RPs with ERG in a single clause, one should thus postulate that this case marks constituents that can be non-unique in a single clause. So far, no such situation has been attested for causative constructions. A thorough investigation of valency increase in languages with different types of syntactic ergativity is definitely needed for the further advancement of this reasoning.

6.2.2.3.4.4 An RRG analysis using oblique core arguments Since analyzing Trumai causatives as simple clauses with three direct core arguments fails, one needs to adopt a different approach. In this section, I suggest treating only ABS and ERG RPs as *direct core arguments*, and DAT – as an *oblique core argument*.

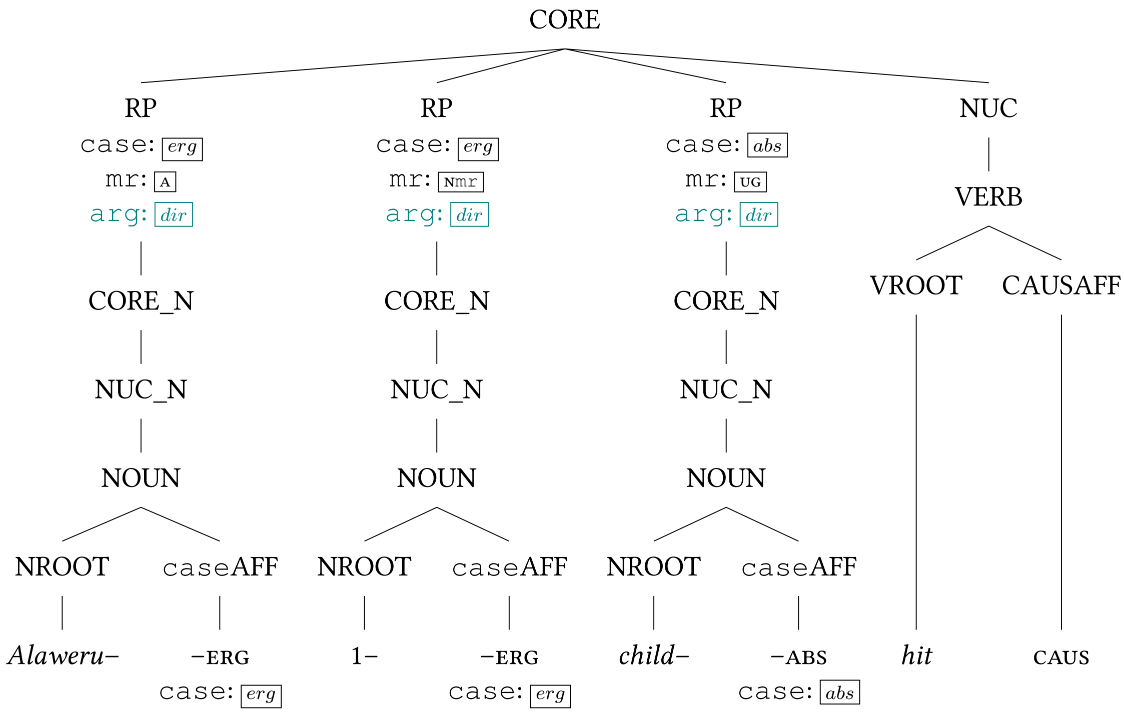
As shown in Section 2.5.1, oblique core arguments are still placed under the CORE node in the tree representation, but cannot bear macroroles. Throughout this section (as well as in most examples in Guirardello-Damian (2003) and Monod Becquelin and Becquey (2012)), dative RPs are different from ergative and absolutive ones as they occur in postverbal positions. Both ERG and ABS are always placed before the verb. This hints that DAT RPs might have a different status. The same sources report, though, that in real texts (in contrast to elicited data), both ERG and DAT can appear preverbally or postverbally.

I thus claim that in Trumai, all direct core arguments that do not bear the UG macrorole receive ERG marking. Oblique core arguments receive DAT, and adjuncts receive other cases. Thus, one can postulate a syncretic ERG case used for different types of arguments able to be marked with different cases in other languages[29].

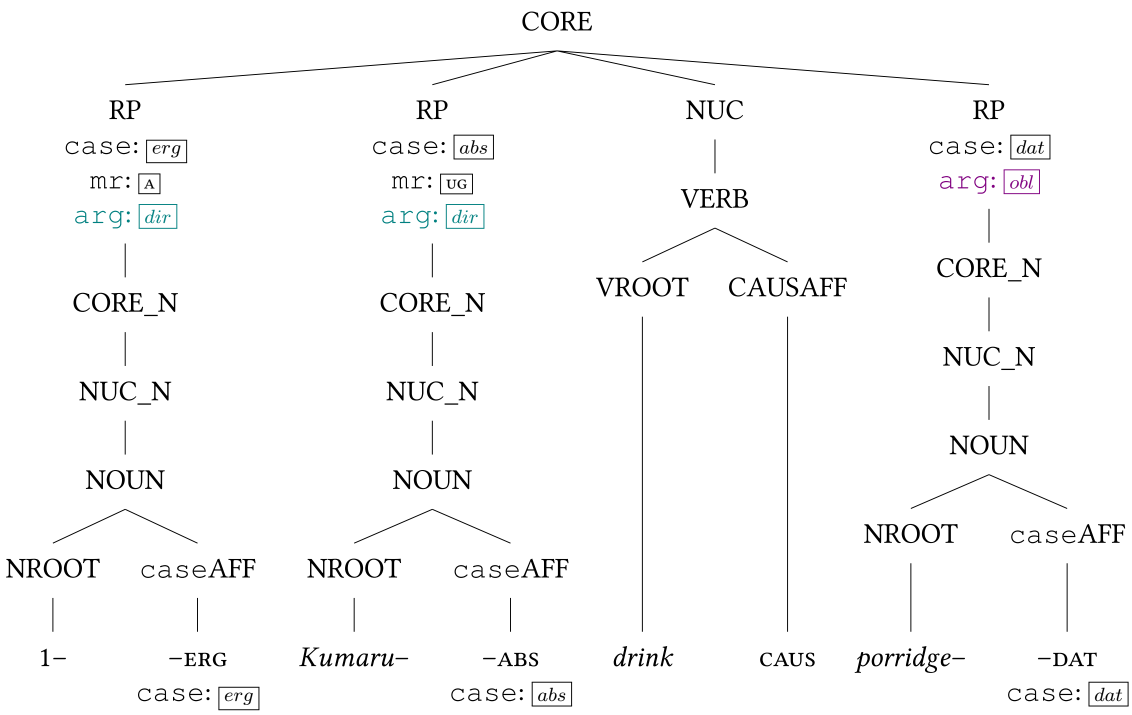
Within this line of analysis, the causativization of extended intransitives (104) follows Comrie's paradigm case: the position for an ERG RP is free in the base verb's alignment and gets filled by the causer. Causatives of transitives like (97)-(98) thus could be explained using Comrie's paradigm case as well: the former Actor (the causee) gets demoted to the position of a NMR core argument, which **also** receives ERG case due to case syncretism, and the position of the Actor gets occupied by the intervening causer and naturally receives its ERG case. The difference from Comrie's paradigm case, though, is that this operation does not yield the same argument pattern as found in non-derived ditransitives (unlike in other languages that demonstrate clear examples of Comrie's idea). This has not been attested so far, but, in fact, has nothing to do with causative constructions. The peculiarity of Trumai ditransitives can be explained via rules of DAT assignment. The fact that DAT is used for oblique core arguments but not for direct ones is a special feature of the Trumai language.

To sum up, I suggest an analysis of Trumai causative constructions as simple clauses, where the duplication of ergative RPs is conditioned by the fact that ERG is a syncretic

[29]See details on RRG analyses of case syncretism in Nakamura (2011) and Nakamura (2021).



a. The syntactic representation of (98) 'Alaweru made me beat the child'.



b. The syntactic representation of (104) 'I made Kumaru drink some manioc porridge'.

Fig. 19. Syntactic descriptions of Trumai causative constructions

case assigned to all non-UG direct core arguments. One can regard that as an "invisible demotion": the agent of the base verb is demoted in semantic terms but receives the same ERG case, so it cannot be distinguished from the other ERG-marked argument on the surface level.

This analysis does not contradict any descriptive evidence, making it preferable to all previously suggested ones. Nevertheless, it suggests an 'invisible demotion' in causatives-of-transitives, which might be an over-complication. Despite this known imperfection, I use this analysis as final for the time being.

Syntactic trees for Trumai causative sentences (98) and (104) are shown in Fig. 19. Note that both of them represent a single CORE (like in all languages examined earlier in this work). Arguments in ERG or ABS are direct core arguments (note the feature `arg`: $\boxed{dir}$) and therefore are also assigned macroroles (see feature `mr`). In Fig. 19b, the DAT argument is still a core argument (consider the same edge from CORE to RP), but is oblique (note the feature `arg`: $\boxed{obl}$). For this reason, it is not assigned any macrorole, hence the lack of the feature.

6.2.2.3.5 Conclusion This section has shown that Trumai is, in fact, a challenging language for the study of causative constructions. Some inspirational news, though, is that more studies of basic alignment patterns could advance research on more complex issues in argument structure. I have explored a line of analysis that assumes Trumai has some syntactic ergativity, but it failed to explain the lack of demotion and the doubling of ERG cases without making other, even less well-supported assumptions. This direction seems the most urgent for a better understanding of Trumai argument patterns and the place of Trumai in the typology of the world's languages. I have also explored an approach that treats morphological causatives as CORE junctures, which has been theoretically predicted to be improbable. Although this analysis explains the argument marking, it contradicts the description of Trumai, requiring a different structure of CORE junctures. Finally, I have come up with an application of Comrie's paradigm case to Trumai causatives, postulating that ERG is a syncretic case assigned to all non-UG direct core arguments, and DAT – to oblique core arguments. Due to case syncretism, one faces an 'invisible demotion' in causatives-of-transitives.

If one seeks to relate the causative-of-transitive construction to any other construction in Trumai, it is still unique. The mechanism for deriving it is the same as the mechanism for deriving causatives from extended intransitive and from simple ditransitives as well. I would then suggest calling this general strategy **No visible demotion** and listing it as a separate case in my enumeration.

6.2.2.4 Final remarks on "double subject" strategies

As it follows from the section above, there is no proof of double agents in either Kabardian or Trumai. A closer look into both head-marking and dependent-marking properties of Kabardian (and its closest relative, Agyghe) reveals that several types of constructions can have multiple RPs with the so-called ERG marking. In fact, Kabardian causatives-of-transitive belong to the type **Causee like Recipient**. Trumai appears to be more complex, to some extent, due to a lack of studies. Nevertheless, I have developed an analysis that does not contradict any descriptive evidence and places Trumai in a distinct group. What also needs to be highlighted is that both Kabardian and Trumai have classes of bivalent intransitive verbs. In all these patterns, the position where the Ergative would appear is free; thus, the derivation is more similar to the derivation of the usual intransitive verbs rather than to the derivation from transitive bases.

6.2.3 Strategies with differential object marking

Section 6.2.1 and Section 6.2.2.4 have studied causative strategies involving overt coding conflict, i.e., with some case markers appearing in the sentence twice. In the languages of the world, there is one more causative strategy with partial doubling (or *quasi-doubling*) of objects. This special pattern is attested in languages allowing differential object marking, i.e., where direct objects can *optionally* bear an overt accusative marker. This section aims to examine these strategies across several Altaic languages (see a note on this term in Section 6.1.1.4.2).

The general causative construction with *quasi-doubling* comprises one participant (causee) with explicit accusative marking and another participant (affectee) in an unmarked accusative case. The causer is in the nominative case, which is typically unmarked. In my language sample, there are no ergative languages with this type of construction.

6.2.3.1 Data: Kalmyk, Khakas, Tuvinian

Consider a Kalmyk example (110) illustrating this general pattern. According to Say (2009, 411), the affectee in this type of Kalmyk structures never receives an explicit accusative marker.

(110)	bagšə	madn-igə	škol-də	kögǯmə	soŋs-ul-na	KALMYK
	teacher	we-ACC	school-DAT	music	listen-CAUS-PRS	

‘At school, the teacher made us listen to music.’ Say 2009, p. 411, (36)

Say (2009, 411) notes that constructions with two accusatives are not encountered anywhere else in Kalmyk. Moreover, in these constructions, only a limited number of affectees is found: they are always inanimate, often cognate to the verb, and sometimes they do not add much information to the sentence (what else can one listen to, if not music?), or are idiomatically bound to the verb (Say 2009, 411). The author comes to the following conclusion:

> Apparently, the combination of these objects with the base verb constitutes a single inseparable predicate and not a full VP. Therefore, under causativization, they behave like intransitive verbs. Say (2009, 412)

Another language with such a pattern is Khakas (Turkic). The situation in Khakas is similar but less strict regarding double explicit accusative marking. Letuchiy (2006, 422) lists pattern (111) as productive, whereas examples like (112) are mostly prohibited and judged by some informants only as "possible, though highly improbable" (Letuchiy 2006, 421).

(111) Maša Vas'a-ny sok iz-irt-če — KHAKAS
Masha Vasja-ACC juice drink-CAUS-PRS
'Masha gives Vasja juice to drink.' — Letuchiy 2006, p. 422, (11b)

(112) ??Pastax toɣasčy-n mašyna-ny š'un-dur-š'e — KHAKAS
boss worker-ACC car-ACC wash-CAUS-PST
'The boss made the worker wash his car.' — Letuchiy 2006, p. 421, (9)

For completeness, I would like to list one more language, namely, Tuvinian (also known as Tuvan, Turkic). As for Tuvinian, Kulikov (1998) lists both strategies: with two explicit accusative markers (113) and with the unmarked affectee (114). Note that the translations differ in the definiteness of the affectee.

(113) Ašak Bayïr-nï inek-ti oorla-t-kan. — TUVINIAN
old.man Bayir-ACC cow-ACC steal-CAUS-PST
'The old man made Bayir steal the cow.' — Kulikov 1998, p. 261, (11c)

(114) Ašak Bayïr-nï inek oorla-t-kan. — TUVINIAN
old.man Bayir-ACC cow steal-CAUS-PST
'The old man made Bayir steal a cow.' — Kulikov 1998, p. 262, (13b)

The status of the causative strategy with two explicit causative markers (113) in Tuvinian is not quite clear. First, Kulikov (1998, 261) tells that it is judged just as "less acceptable" than the strategy with the dative causee, but afterward, he makes a stronger claim: "it is worth emphasizing that such constructions with two accusatives are not appropriate for Tuvinian nor for many other Turkic languages". Commenting on constructions of type (114), Kulikov (1998, 261–262) notes that "the accusative marking of the causee becomes preferable (or even obligatory) in a special situation, namely when the embedded direct object appears in the so-called non-marked accusative form".

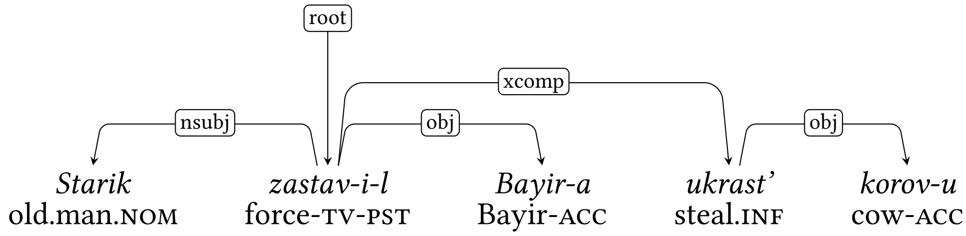


Fig. 20. A dependency syntactic analysis of the Russian causative stimulus (115)

Thus, I suggest examining three genetically related languages, each with a causative strategy that uses the accusative causee and the affectee in an unmarked accusative case. Interestingly, these languages differ in their acceptability of the causative construction with two overt accusative markers. Below, I present prior analyses of these phenomena from descriptive studies.

6.2.3.2 Prior analyses: language contact

Both Letuchiy (2006, 421) (for Khakas) and Kulikov (1998, 261) (for Tuvinian) agree that constructions with two explicit accusative markers seem to emerge due to the influence of Russian, a language that is spoken by a significant number of Khakas and Tuvinian speakers (and, importantly, also the language of the elicitation stimuli). As Russian lacks a morphological causative, the basic way to express the same meaning is a biclausal construction with some verbs with a causative meaning as a finite predicate and the verb denoting the caused event as an infinitive. Russian also lacks a verb that has causative semantics only (cf. English *make*), so fieldwork researchers tend to use the verb *zastavljat'/zastavit'* 'to force', see (115) The two accusatives in (115) are objects of the two different verbs. For a better illustration, consider the dependency graph in Fig. 20.

(115) Starik zastav-i-l Bayir-a ukrast' korov-u — Russian
old.man.NOM force-TV-PST Bayir-ACC steal.INF cow-ACC
'The old man forced Bayir to steal a/the cow.' — Kulikov 1998, p. 261, (12)

In this thesis, I am not dealing with the effects that language contact might have on argument marking in causative constructions (although there are works that draw special attention to this, e.g., Luraghi and Roma (2021)). Therefore, I do not analyze this contact-induced pattern any further and do not include it in the enumeration of the causative strategies I make. It is definitely preposterous to draw similarities between examples like (112) and (113) on the one hand, and Bantu-like double object constructions, on the other hand.

However, the language contact hypothesis does not explain the existence of Altaic causatives with unmarked affectees. Hence, a different analysis is required.

6.2.3.3 Prior analyses: incorporation

As already shown in Section 6.2.3.1, Say (2009, 411–412) suggests that Kalmyk structures like (110) have the same argument structure as the causatives-of-intransitives.

Kulikov (1998, 262) puts this argument in an even stronger way. First, he reminds the reader that referential properties are of exceptional relevance for objects in Turkic languages, so the lack thereof is an important indicator of a "pseudo-DO", i.e., a syntactic unit that looks like a direct object but is not. He then supports the incorporation analysis, claiming that "such a noun phrase displays a closer affinity with incorporated nouns so that these constructions might be interpreted as an example of analytical incorporation".

Letuchiy (2006, 431) also attests a closer boundary between the "initial object" (= patient of the base verb = affectee of the causative construction) and the verb. However, in contrast to Say (2009) and Kulikov (1998), Letuchiy (2006) treats it as evidence for biclausality rather than incorporation. With this analysis, the causer, the causee, and the predicate CAUS (apparently, introduced by the affix) are claimed to belong to the external clause, while the affectee and the base verb form the internal clause.

Although the biclausal analysis seems to be more complicated than the incorporation analysis, it reminds us of a similar debate concerning Japanese constructions that are morphological but act as analytical (see Toratani (2002) for the details). Even though these analyses explain well only some properties of the constructions in question, the recurrence of this debate seems to indicate an understudied zone in verbal derivation. Perhaps, a thorough analysis of constructions, now referred to as "partially bicalusal", is yet to come.

Another motivation for the biclausal analysis suggested by Letuchiy (2006) might be the influence of language contact (especially with Russian), which has already been presented and needs further development beyond the present thesis.

6.2.3.4 An RRG analysis

As shown earlier in Fig. 15, RRG allows incorporated nouns to become part of the NUC. In Van Valin and LaPolla (1997, 64–67), options of even tighter compounding are discussed, but they are definitely not relevant for structures like (114), where the incorporation is not complete. Hence, the placement of the unmarked object under the NUC node seems to be the most relevant analysis.

This analysis is shown in Fig. 21. The NP is placed below the NUC node. It means that all layers of the NP structure, namely CORE_N and NUC_N, are also created below the NUC node. This structure tells that the noun *inek* 'cow' can have its own adjuncts and operators. Presumably, the availability of those for an "incorporated" noun is limited. In this case, some of the layers should not appear in the final tree. However, the determination of such properties is language-specific and lies beyond the scope of the

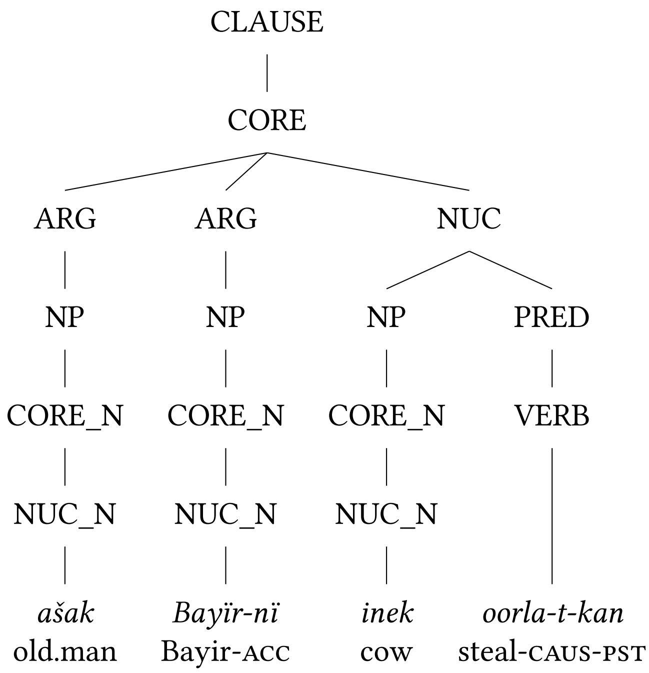


Fig. 21. The tree representation of the Tuvinian causative with an "incorporated" affectee (114)

present thesis. Therefore, I suggest this general analysis as a causative strategy **Affectee incorporated** without further precision of the details of this incorporation in each of the languages.

The crucial question for the current study is the place of the "incorporation". Does it happen before the causative derivation or together with it? In other words, is the incorporation of the affectee a method of NP density control, in Song's terms (Song 1991), or an independent process? The remark by Say (2009) that the argument pattern found in (110) is not attested in any other Kalmyk construction hints that the incorporation might be due to causativization. Nevertheless, a special investigation of the "predicates" built by a verb and an incorporated affectee in other contexts is required as a further research step.

6.2.3.5 Summary

In this section, I have shown a strategy of NP density control found in some Turkic (Altaic) languages that leads to the "incorporation" of the affectee to the "verbal complex". As a result, the causative construction with a transitive base verb has the argument structure of the causative construction with an intransitive base verb. Namely, the causer has NOM marking, the causee has ACC marking, and the affectee remains unmarked since it is incorporated and loses some of its properties. After comparing three genetically related languages, I have argued that the strategy with the "incorporated" affectee is unrelated to the strategy with double accusative marking, which, in turn, seems to emerge from language contact with Russian. I offer a generalized analysis of noun incorporation, leaving language-specific details of its realization for further studies.

Chapter 7

From individual analyses to constructional schemas

In the previous chapter, Chapter 6, I have offered RRG analyses of causative constructions in individual languages and determined to what other constructions they can be related.

In this chapter, I first summarize my findings for individual languages and then offer cross-linguistic generalizations from my analyses. I formulate these generalizations as Constructional Schemas (introduced in Section 2.7). Thus, for each construction, I provide the scheme of argument encoding and of macrorole assignment. I do that for all strategies determined in Chapter 6, be they predicted by the standard RRG linking or not. In this respect, I consider "standard linking" to be just one type of construction, described in the same way as "non-standard" constructions that deviate from it.

This equal treatment of all strategies proves especially useful for implementing my solution. What I actually model in Chapter 10 are these constructional schemas, see details in Section 10.4.5.

7.1 Summary of observations concerning individual languages

One of the aims of Chapter 6 is to update the information on morphological causatives derived from transitive bases for each of the languages I engaged with. The main contribution of the present work is to reconcile descriptive studies of individual languages with prior theoretical studies from diverse approaches, and to offer an RRG-inspired analysis that does not contradict empirical observations. This is made possible due to the method of relating causative constructions to other constructions in the same language (not necessarily those from which causatives are derived) in terms of comparative concepts (as defined in Haspelmath (2010)).

Using this approach, I have offered a more precise analysis for languages, some of which were challenging for prior studies.

Namely, the method relating causatives to other constructions permits regarding Nivkh causatives as a variant of a more widespread strategy **Causee like Demoted Agent**. Before, Nivkh causatives were believed to be unique and non-relatable to any other constructions. Nevertheless, the data I show suggest that the suffix -*aχ* appearing on the causees occurs only in clauses, where the verb has the marker *-gu-* (or another marker that is diachronically related to it). Further Nivkh studies might wish to reconsider the case status of the suffix -*aχ*.

Another commonly problematic causative construction has been attested in Javanese. Calling recent corpus studies on Javanese, I claim that this construction does not exist in modern standard Javanese but might nevertheless survive in some other varieties. Javanese demonstrates morphological syncretism in ditransitive, causative-of-transitive, and applicative-of-transitive constructions: all of them are built with the same verbal marker and allow identical argument-marking strategies. Without studying this syncretism in more detail, it is preposterous to claim the existence of any morphological causative-of-transitive construction in Javanese at all.

Kabardian three-argument causatives have been considered challenging for having two participants looking like the syntactic subject. Thus, a fair amount of reflection was given to postulating some bi-eventive structures in them. Prior studies, however, have overlooked that Kabardian has not only case marking on its arguments but also a well-developed system of verbal prefixes cross-referencing the arguments. With due attention to these head-marking properties and a closer look into the functions of each of the case markers available in Kabardian, I claim that causatives-of-transitives in Kabardian demonstrate the same argument-marking strategy as underived ditransitives.

Another language suspected to have double subjects in causative constructions, Trumai, appears, in fact, quite challenging. Its basic alignment patterns are yet unclear, hence, relating causatives to any other constructions is quite hypothetical. Nevertheless, I manage to trace some generalizable correspondences between attestable patterns. I claim Trumai to belong to a wide group of languages where causatives-of-transitives follow a ditransitive argument pattern. This claim holds, though, if one postulates some case syncretism in Trumai, for which there is still descriptive evidence lacking. I also outline several lines of analysis I failed to pursue to their conclusions, providing directions for possible future studies of Trumai alignments.

Another language I re-classified is Finnish. Previously, it has been believed to have a rare type of locative causees. I have demonstrated that the ADESS causee marking is also found in constructions with prototypical instruments and thus claimed that the causative strategy can be called **Causee like Instrument**.

Regarding instrumental marking, I have shown that it should be regarded as three distinct argument-marking strategies. I have found three types of languages. First, those that mark causees like instruments but not like demoted agents in passive clauses, namely Finnish. Second, those that mark causees like demoted agents in passive clauses, but not like instruments, namely Bashkir. Third, those that use a single marking for instruments, demoted agents, and causees. The syntactic properties of arguments in these three strategies differ; therefore, it is especially important to differentiate among them. Quantitative typological studies should also account for this difference if they aim to establish how frequently each strategy occurs.

I have also dealt with the phenomena of object (a)symmetry and locus of marking in Bantu languages. The Lubukusu data suggest that not languages as a whole but individual constructions have to be classified as symmetrical or asymmetrical. The analysis departing from Ackerman, Malouf, and Moore (2017) allowed me to treat symmetry and asymmetry as equally possible variants and thus offer a novel analysis of Zulu and Lubukusu. In Zulu and Lubukusu, the causative derivation does not affect the M-transitivity. The difference between M-transitivity and S-transitivity greater than one is allowed. In Lubukusu, verbs of ingestion are M-intransitive. Once causativized, three-argument M-intransitive verbs have two postverbal object-like NPs, neither of which is assigned a macrorole. This results in their symmetrical behavior. In Zulu, all verbs behave like that. In contrast, non-ingestive verbs in Lubukusu do not allow three direct core arguments. When causativized, they exhibit a gradual demotion of all the participants: the causer occupies the psa position and bears the Actor macrorole, the former agent (i.e., the causee) becomes the direct core non-macrorole argument (able to take part in syntactic processes), finally, the former patient is demoted to the position of an oblique core argument (not eligible for syntactic processes). The same pattern is also attested to instrumental and locative applicatives in Lubukusu.

As for the locus of marking, I examined data from Kinyarwanda and Kikuyu. I partly agree with the analysis originating in Bresnan and Mchombo (1987) and presented in Kihara (2017) in the treatment of object markers: I suggest analyzing them as incorporated pronouns. In contrast, I disagree with these studies' approach to subject markers and offer an analysis of Kinyarwanda and Kikuyu similar to that of dependent-marking prodrop languages.

Finally, I drew attention to a peculiar construction of double accusative marking in Kalmyk, Tuvinian, and Khakas. I offered an RRG analysis of it, relying on the noun incorporation. I claim that this is one of the methods of NP density control (Song 1991).

7.2 Language profiles

In Tab. 24, I offer a summary of language profiles in relation to the causative constructions they have. For each language, I name the basic (most often unmarked) strategy used in causatives of transitive verbs. In addition, I provide information on whether languages use different means for encoding causatives of ingestives and/or causatives of transfer verbs. I also list an option for the choice of an argument pattern being pragmatically motivated, and, if so, which strategies occur in languages that are sensitive to that. Not surprisingly, some parameters are often irrelevant for a given language, as my analysis also shows. Quantitative typological studies could benefit from taking the relevance of each parameter into account. For some cells, I admit I was unable to find sufficient evidence to assign a specific value to those cells (see '*no data*').

These profiles are used in my implementation, as demonstrated in Chapter 10, to parse sentences in individual languages correctly.

Language	Basic construction	Any special construction for ingestive predicates?	Any special construction for transfer predicates?	Any effects of the Agentivity Implicature?
Kannada	Causee like Recipient	no	no	explicit favoring: Causee like DemAg / Instr (syncretic)
Kabardian	Causee like Recipient	no	no	no
Javanese	Causee like Recipient	(no data)	no causative	(no data)
Bashkir	Causee like DemAg 1	Causee like Recipient	Causee like Recipient	no
Nivkh	Causee like DemAg 2	no	no	no -AX marking (needs proof)
Finnish	Causee like Instrument	(no data)	(no data)	(no data)
Zulu	Causee and Affectee like Direct Object	no	no	no
Lubukusu	Causee like Direct Object, Affectee like Pseudo Object	suspected non-derived ditransitives	(no data)	no
Trumai	Causative with no demotion	no	no	no
Tuvinian	Affectee incorporated OR Causee like Recipient	(no data)	(no data)	no

Table 24: The range of causative constructions attested in sample languages

7.3 Summary of observations concerning cross-linguistic generalizations

When analyzing individual languages, I have observed some similar trends. The first trend is that, in many languages, causatives derived from ingestive transitive verbs behave differently from the causatives derived from other verbs. Namely, in Lubukusu, causatives-of-ingestives are symmetrical, while other causatives-of-transitives are asymmetrical. In Bashkir and some other Altaic languages, causatives-of-ingestives tend to take dative causees Hence, it is clear that the semantics of ingestion, or its event structure, is crucial for the perception of causality across languages. One possible explanation is that the base structure of verbs of ingestion differs from that of more prototypical transitive verbs. Another explanation for the selection of a construction from a range of possibilities is that causatives of ingestives are more closely related to monotransitive verbs than causatives of other verb classes. Further, typological studies should take this cross-linguistic trend into account.

Another class of verbs that cross-linguistically tend to require a special causative construction are verbs of physical or non-physical transfer. These verbs have a semantic component that is also present in prototypical ditransitive constructions. In many cases, it is difficult to tell the difference between ditransitives and such causatives, since they can have the same set of arguments and identical verbal morphology. Such constructions can be regarded as syncretic. A broader view of three-participant constructions, combined with a thorough analysis of each marker, could help define the boundaries and the extent of such syncretic patterns in individual languages, as well as formulate precise cross-linguistic generalizations.

Another trend, well attested previously, is the co-occurrence of several causative constructions in a single language. I have engaged with the debate and found that the axis of the participant's volitionality cuts across the axis of the construction's implicativeness (non-defeasibility), as both play a role in an analysis of a given sentence.

This revealed the confusion arising from several strategies; two functional positions – a prototypical instrument and a demoted agent – have obscured prior analyses. Languages appear to differ in this respect: some of them mark causees like instruments, and others mark causees like demoted agents. Many languages also have a single syncretic marker for all these positions.

Comparing causees that look like recipients of ditransitives and demoted agents of passives, I expand the RRG analysis of this competition from the perspective of discourse/pragmatics. I show that the previously formulated Agentivity Implicature has an effect only in cases of non-defeasible causation and seems to play no role in cases of defeasible causation.

When studying the similarity of causative constructions to prototypical ditransitive constructions, I suggested a systematic relationship between ditransitive and causative alignments:

(116) a. If both – the ditransitive and the causative – constructions follow *the same* alignment pattern (either indirective or secundative), the causee receives the same marking as the *recipient* of a ditransitive.

b. If both – the ditransitive and the causative – constructions follow *different* alignment patterns, two options are possible: (1) the causee receives the same marking as the *theme* of a ditransitive OR (2) the *affectee* receives the same marking as the recipient of a ditransitive.

This conclusion answers the question of why constructions of type **Causee like Recipient** are extremely more frequent than the other two types. First, they are attested in languages with any ditransitive alignment (while each of the strategies **Causee like Theme** and **Affectee like Recipient** is attested in languages with one type of ditransitive alignment only). Second, and most importantly, it is not very plausible that a language would use different kinds of alignment for different three-argument constructions. Reasons that might lead to this are a topic for upcoming research in the typology of alignment.

7.4 Constructional schemas

One of the aims of Chapter 6 was to offer an enumeration of strategies used for marking arguments in constructions with morphological causatives derived from transitive bases. In Tab. 25, I summarize my findings, offering an updated set of constructional schemas that are comparable to one another. Like traditional constructional schemas, they list properties of a construction belonging to every relevant level of the language. The field `formula` tells how morphologically the participants of the causative construction must be encoded. The general indices *i, j, k* refer not to hard-coded cases or other ways of marking but to ways of marking used in some non-causative constructions. The coding means used in each language are not part of the cross-linguistic generalization and, thus, are not included in the constructional schema. The fields `arguments` and `macroroles` provide a formulaic representation of the values that the two most important features assigned to each participant have.

Strategy	Relatable construction	Argument encoding	Macrorole assignment	Example
Causee like Recipient	ditransitive	$E_i^{caus}\, C_j^{caus}\, A_k^{caus}\ :\ A_i^{ditr}\, R_j^{ditr}\, T_k^{ditr}$	$E_a^{caus}\, C_{nmr}^{caus}\, A_{ug}^{caus}$	Kannada (36a)
Causee like Instrument	instrumental	$E_i^{caus}\, C_j^{caus}\, A_k^{caus}\ :\ A_i^{instr}\, I_j^{instr}\, P_k^{instr}$	$E_a^{caus}\, C_{nmr}^{caus}\, A_{ug}^{caus}$	Finnish (57)
Causee like Demoted Agent 1	passive	$E_i^{caus}\, C_j^{caus}\, A_k^{caus}\ :\ A_i^{tr}\, U_k^{tr}\ :\ demA_j^{pass}$	$E_a^{caus}\, C_{nmr}^{caus}\, A_{ug}^{caus}$	Bashkir (38b)
Causee like Demoted Agent 2	reported speech	$E_i^{caus}\, C_j^{caus}\, A_k^{caus}\ :\ A_i^{tr}\, U_k^{tr}\ :\ demA_j^{rep}$	$E_a^{caus}\, C_{nmr}^{caus}\, A_{ug}^{caus}$	Nivkh (63)
Causee and Affectee like Direct Object	simple transitive	$E_i^{caus}\, C_j^{caus}\, A_j^{caus}\ :\ A_i^{tr}\, U_j^{tr}$	E_a^{caus}	Zulu (77a)
Causee like DO, Affectee like Pseudo Object	simple transitive	$E_i^{caus}\, C_j^{caus}\, A_j^{caus}\ :\ A_i^{tr}\, U_j^{tr}$	$E_a^{caus}\, C_{ug}^{caus}$	Lubukusu (72)
Affectee incorporated	simple transitive	$E_i^{caus}\, C_j^{caus}\, (A_{\varnothing}^{caus})\ :\ A_i^{tr}\, U_j^{tr}$	$E_a^{caus}\, C_{ug}^{caus}$	Tuvinian (114)
Causative with no demotion	simple transitive	$E_k^{caus}\, C_i^{caus}\, A_j^{caus}\ :\ A_i^{tr}\, U_j^{tr}$	$E_a^{caus}\, C_{nmr}^{caus}\, A_{ug}^{caus}$	Trumai (97)

Table 25: Constructional schemas for causative constructions

Part III

Implementation

Chapter 8

From theoretical abstractions to tangible representations

8.1 Introduction and motivation

The motivation to develop computer-compatible representations of knowledge is increasingly recognized as essential in contemporary research. In linguistics, a growing number of resources are designed for machine processing rather than human use. With the aid of well-crafted software, machines can execute repetitive tasks—such as parsing sentences—more rapidly and with greater accuracy than human counterparts. Furthermore, creating a robust implementation for newly introduced data poses its own set of challenges, serving as a rigorous evaluation of the proposed analytical frameworks. This underscores the necessity of integrating computational methods into linguistic research, as they not only enhance efficiency but also deepen our understanding of linguistic phenomena.

In this line, I cannot leave my findings just as tables shown in Chapter 7 but should provide a more solid computational implementation. Thus, the implementation will provide correct parses of the sentences I engaged with in Part II and justify some decisions I made for my analyses from the logical and computational perspective.

Another advantage of implementing theoretical analyses is their standardization, which entails compatibility with a larger community. I argue that the solution I present in Chapter 10 can be easily and quickly expanded to include more languages and more linguistic phenomena. Once again, the implementation could yield new insights for the theory, offering observations on which fragments of the analyses can be repeated, generalized, and transferred.

8.2 From classical RRG to a formalized tree-wrapping grammar

8.2.1 Background

Classical RRG is not a formal grammar. It often formulates its claims as commands that should be applied by someone while working with the data. The Linking Algorithm is very illustrative in this respect, being basically a set of imperative instructions. Consider its beginning in Van Valin (2005, (5.10), p. 149): "Determine the macrorole(s) and other core argument(s) in the clause. If the verb is intransitive, then assign the privileged syntactic argument either macrorole or direct core argument status, depending upon the language (language-specific)".

These instructions can, in fact, be fed to a computer. This step inevitably involves some modification of the classical theory since machines work best with *formalized* knowledge. The formalization of RRG syntax and semantics has been suggested in Osswald and Kallmeyer (2018), and the newest version of the syntax formalization recently appeared in Kallmeyer and Osswald (2023). The authors set two main goals: "A formalization can, however, help to identify and eliminate possible gaps and inconsistencies of the theory and, thereby, to improve the theory. Moreover, a formalization can serve as a basis for computational implementations of RRG" (Kallmeyer and Osswald 2023, 738).

When formalizing RRG, the authors suggest keeping the tree structures as the general method of syntactic representation. There already exists multiple grammar formalisms for describing trees, one of which is TAG (Tree-Adjoining Grammars, Joshi and Schabes (1997)). Note that a grammar formalism is different from a linguistic theory in that it does not make any claims about the contents of the trees but regulates the way they are composed. Thus, the TAG formalism is initially defined as a "quintuple" of the following elements, as shown below.

> A tree-adjoining grammar (TAG) consists of a quintuple (Σ, NT, I, A, S), where
>
> 1. Σ is a finite set of terminal symbols;
> 2. NT is a finite set of non-terminal symbols: $\Sigma \cap NT = \varnothing$;
> 3. S is a distinguished non-terminal symbol: $S \in NT$;
> 4. I is a finite set of finite trees, called *initial trees*, characterized as follows <...>:
>   - interior nodes are labeled by non-terminal symbols;
>   - the nodes on the frontier of initial trees are labeled by terminals or non-terminals; non-terminal symbols on the frontier of the

trees in I are marked for substitution; by convention, we annotate nodes to be substituted with a down arrow (↓);

5. A is a finite set of finite trees, called auxiliary trees, characterized as follows <...>:
   - interior nodes are labeled by non-terminal symbols;
   - the nodes on the frontier of auxiliary trees are labeled by terminal symbols or non-terminal symbols. Non-terminal symbol on the frontier of the trees in A are marked for substitution except for one node, called the foot node; by convention, we annotate the foot node with an asterisk (*); the label of the foot node must be identical to the label of the root node.

Joshi and Schabes (1997, 70–71)

Quite often, the requirement of lexicalization is applied to TAG trees: "at least one terminal symbol (the anchor) must appear at the frontier of all initial or auxiliary trees" (Joshi and Schabes 1997, 72). There are also two operations defined for combining these trees: *adjunction* and *substitution*.

Without going into much detail about the TAG formalism, one can note that it is compatible with various linguistic analyses. Thus, the task of formalizing RRG as a variant of TAG primarily involves identifying the elements that conform to the formal TAG definition in classical RRG.

Nevertheless, the translation is not straightforward. The set of operations defined in TAG aims at modifying binary branching structures (Joshi and Schabes 1997, 103), while RRG trees often comprise non-binary branching. Hence, the innovation of Kallmeyer, Osswald, and Van Valin (2013) and Osswald and Kallmeyer (2018) is in defining novel parsing operations that best account for RRG-like trees. The new formalism is called TWG, which stands for Tree-Wrapping Grammars, since *wrapping* is one of those newly defined operations (Kallmeyer, Osswald, and Van Valin 2013).

8.2.2 Operations in TWG

Instead of simple substitution and adjunction, TWG suggests three operations: (simple) *substitution*, *sister adjunction*, and *wrapping substitution*. These three operations are exemplified in Fig. 22.

The operation of *substitution* requires a tree with at least one leaf node ("frontier" in terms of Joshi and Schabes (1997)) marked as available for substitution (often referred to as a *substitution node*). Historically (following Joshi and Schabes (1997)), the '↓' sign is used for that. The substitution node can unify only with a node of the same category, namely, the root (highest) node of a different tree. After substitution, a new tree is

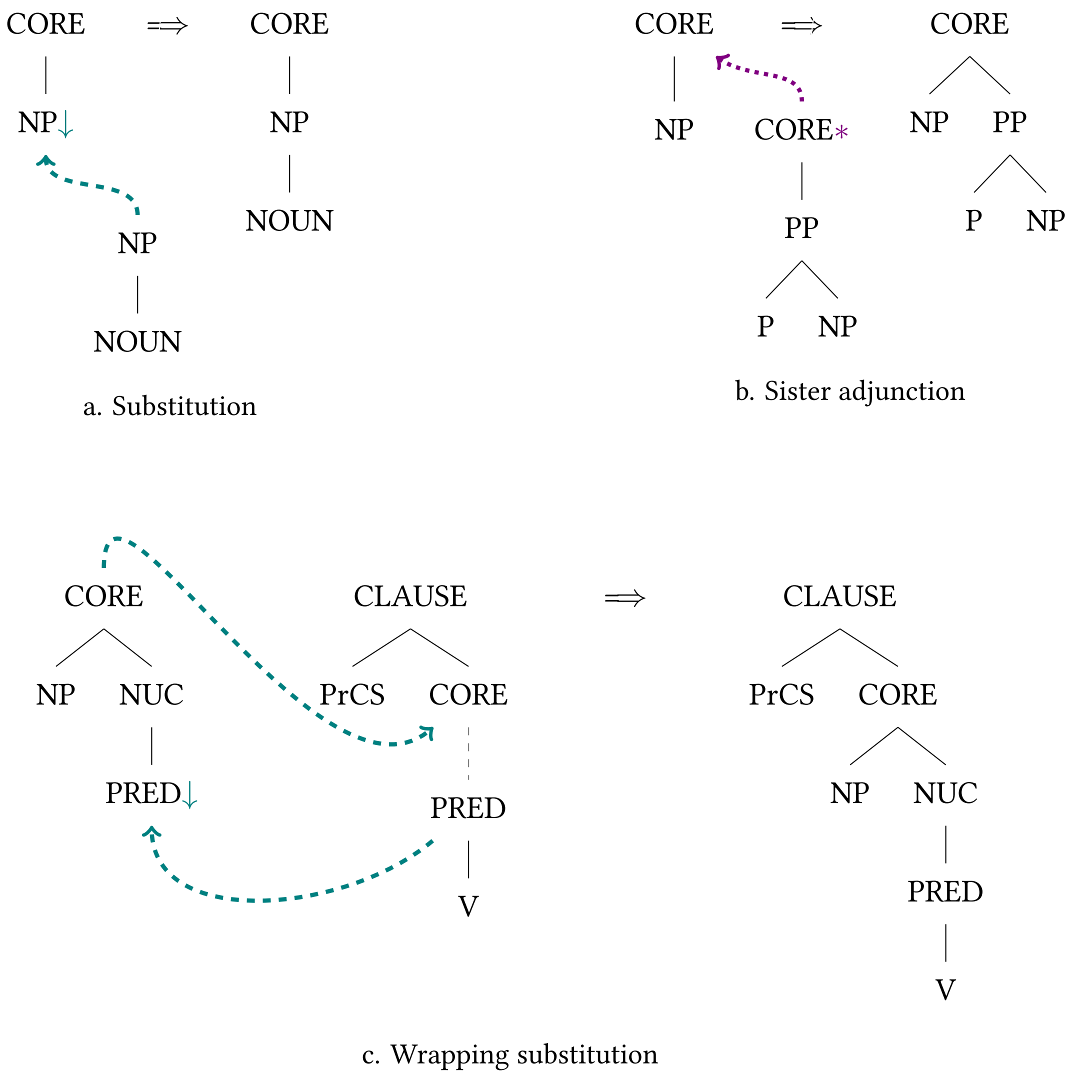


Fig. 22. Parsing operations (reproduced from Kallmeyer and Osswald 2023, Fig. 20.7, Fig. 20.11, modified, simplified)

produced, the upper part of it coming from the tree with a substitution node and the lower part coming from the substituting tree. In Fig. 22a, the node NP↓ can be substituted by any tree rooted in an NP node. One of such trees is also shown in Fig. 22a, and the operation of substitution is illustrated with a dashed line. The resulting tree has a single NP node since the substitution node and the root node of the substituting tree unify.

The *sister adjunction* is a flexible operation that allows for adding daughter subtrees to a selected node of a syntactic tree (or, in RRG terms, to a selected layer of the clause). Contrary to the substitution, the special mark '∗' is placed not in the upper tree (*to which* a new subtree adjoins) but in the *adjoining* (also referred to as *adjunct*) tree. The structure of the adjunct trees is strictly defined. First, their root must be of the same category as the category to which the tree is supposed to be adjoined. Secondly, the root of the adjoint tree must have only one daughter (Kallmeyer and Osswald 2023, 747). Otherwise, after the unification of the adjunct root with the respective node of the upper tree, additional instructions on the ordering of the incoming subtrees would be needed. The sister adjunction, however, does not specify the linear place where the adjunct tree should appear, which is extremely helpful for handling RRG structures, especially operators. However, when needed, precedence constraints can be introduced to control the order of the daughter subtrees.

Sister adjunction is illustrated in Fig. 22b with a dotted line. The adjunct tree shows a PP that must attach to the CORE. Note that PP is the only daughter of the CORE∗; however, it may have several child nodes itself. Although in this example, the resulting tree is binary, no such requirement exists in the formalism. Namely, adjunct trees can be added to trees with any number of branches. Moreover, any number of adjunct trees can attach to the same tree.

The third operation is *wrapping substitution*, see Fig. 22c. It includes two substitution operations shown as dashed lines. The idea of this operation is to wrap one tree around another, i.e., split a tree and use two simple substitution operations. The place where the tree is split is indicated by the so-called *d-edge*, shown as a thin dashed edge in Fig. 22c.

Thus, the tree rooted in the CLAUSE node in Fig. 22c functions simultaneously as two types of structures. On the one hand, it is a single tree and thus allows for describing the PrCS and the PRED together. On the other hand, it functions as two trees: the lower part can fill a substitution node of another tree (like the tree rooted in NP in Fig. 22a), while the upper part contains a parent node of the d-edge (CORE in Fig. 22c) that seeks to unify with the root of another tree.

The tree around which a tree with a d-edge can be wrapped is sometimes referred to as the *target tree*. In Fig. 22c, it is the tree rooted in the CORE node. To become an eligible target for a given wrapping tree, a tree must fulfill two conditions. First, its root must be of the same category as the parent of the d-edge in the wrapping tree. Second, one of its lower ("frontier" in terms of Joshi and Schabes (1997)) nodes must be of the same category as the child of the d-edge and bear a substitution mark ↓.

The result of the wrapping is shown in the right-hand side part of Fig. 22c. One can see that the target tree is inserted in the place of the d-edge. The wrapping substitution is extremely useful for describing non-local dependencies, such as *wh*-extraction in English. In my solution, I use it for describing arguments in head-marking languages[30](see Section 10.4.2.4).

Once the classical grammar is formalized, the simplest way to implement it would be to simply list the tree-frame pairs for all the target sentences manually. This approach has multiple inconveniences. First of all, nothing prevents manual errors in these representations. Second, most representations look alike and contain common parts; hence, writing every tree from scratch requires a lot of repetition. Being able to create each common part only once would significantly decrease the time required for project development and the number of errors – at least for repeated parts. Moreover, simply listing all options does not benefit project scaling. The addition of every new target sentence (with previously unseen grammar rules or lexical items) would take the same time and effort. Finally, and importantly for the development of a linguistic theory, the simple listing of representations does not reflect linguistic generalizations.

In fact, there is a solution able to overcome these challenges. The key point is *factorization*, i.e., abstraction. The higher the degree of factorization, the less repetition there is in the project. Splitting tree-frame pairs corresponding to full sentences into TWG elementary trees is already some level of abstraction. However, it is possible to describe elementary trees in an even more abstract way. This approach, together with the tools I use for it, is presented in Section 8.4.

Before continuing with the presentation of the metagrammatical approach in general and the tools for creating metagrammars that I used in my implementation, I would like to present the formalization of RRG semantic structures, also suggested in Osswald and Kallmeyer (2018).

8.2.3 Frame semantics for RRG

Osswald and Kallmeyer (2018, 372) note that the logical structures used for describing semantics in classical RRG "from a formal point of view, are meaningless expressions built in accordance with certain formation rules". The authors also note that "there are a number of additional semantic components used within RRG's logical structures whose formal syntax seems not fully fixed yet". Hence, the authors suggest formalizing the

[30]The idea to use wrapping substitution to describe arguments in head-marking languages was suggested to me by my supervisor, Laura Kallmeyer, and supported by my advisor, Rainer Osswald, during one of the meetings dedicated to the discussion of my doctoral project. The idea was (un)published as a small handout (Kallmeyer 2021) and later modified by me to suit the architecture of my solution.

semantic structures alongside the syntax to enable RRG to parse real language data automatically.

In Osswald and Kallmeyer (2018), when suggesting the formalization of RRG syntax based on TAG, the authors suggest translating RRG's logical structures into *decompositional frames*, which have been used with TAG earlier (Kallmeyer and Osswald 2013). Osswald and Kallmeyer (2018, 372) highlight that this architecture allows "to keep the key properties of RRG's decompositional system without necessarily preserving the specific form of the logical structures" and nevertheless have a formal account suitable for further implementation. The authors show that RRG-like logical structures can be straightforwardly translated into frames, including the type of event, its subcomponents, and the roles and labels of arguments.

In the solution presented in Chapter 10, frames are used and linked to the syntactic representations (see Section 10.4.3 for details). However, the primary focus of my project is the argument structure of causative sentences. Hence, the semantic structures I provide in the suggested solution are far from presenting all the nuances of the meaning of each construction. Instead, they are regarded merely as argument structure templates. For this reason, I mainly reuse the existing way of representing causative frames and do not provide any advanced hierarchy of them. Given that frame semantics is not of primary importance for the presented solution, this section focuses on the mechanism for translating logical structures into frames, presenting it briefly and generally. More details on the formal aspects of frames can be found in other works (Kallmeyer and Osswald 2023; Osswald and Kallmeyer 2018; Kallmeyer and Osswald 2013; Kallmeyer and Osswald 2012; Gardent and Kallmeyer 2003).

In Kallmeyer and Osswald (2013, 278), the authors suggest translating into frames structures offered in Rappaport Hovav and Levin (1998). The approach is illustrated with the decomposition of the verb *break*, see (117). One can see that it is represented as a causative structure.

(117) [x ACT] CAUSE [BECOME [y BROKEN]]]

The structure in (117) can be represented as a causative accomplishment, following Van Valin (2023b, 99). The resulting logical structure is demonstrated in (118).

(118) [**do′** (x, ∅)] CAUSE [BECOME **broken′** (y)]

The CAUSE operator is present in both representations (117) and (118). The first event in (117) is denoted with the keyword ACT, while in RRG, activities are represented as [**do′** (x, ∅)]. The second event is denoted with the keyword BECOME in both structures. Hence, the RRG structure in (118) is equivalent to (117). Therefore, the reasoning applied to (117) by Kallmeyer and Osswald (2013) is also applicable to (118).

Both (117) and (118) are not fully formalized. Namely, the status of the keywords is unclear: do they relate to events, parts of events, or participants? Hence, before trans-

$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \begin{bmatrix}\textit{activity} & \\ \text{ACTOR} & \boxed{1}\end{bmatrix} \\ \text{EFFECT} & \begin{bmatrix} \textit{change-of-state} & \\ \text{RESULT} & \begin{bmatrix}\textit{broken-state} & \\ \text{PATIENT} & \boxed{2}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Fig. 23. The frame represenation of the English verb *break* regarded as a causative (reproduced from Kallmeyer and Osswald 2013, p. 279, Fig. 6a)

lating the structure (117) to a frame, Kallmeyer and Osswald (2013, 279) create an intermediary structure in the Neo-Davidsonian-like style (Parsons 1990), see (119).

(119) causation(e) ∧ CAUSE(e, e′) ∧ EFFECT(e, e″) ∧ activity(e′) ∧ ACTOR(e′, x) ∧ change-of-state(e″) ∧ RESULT(e″, s) ∧ broken-state(s) ∧ PATIENT(s, y)

Note that in (119), all the arguments in brackets are events, subevents, or participants. This precision of the operators finally allows the creation of frames, as shown in Fig. 23. The frame in Fig. 23 has a root (consider label $\boxed{0}$) and two labels for the participants (cf. $\boxed{1}$ and $\boxed{2}$ in Fig. 23 and *x*, *y* in (119)). In Fig. 23, the subevents *e*′ and *e*″ lack special lables. However, they can be added once needed.

To complete the presentation of the frame structures, one needs to specify that they are attribute-value descriptions. A more explicit notation of Fig. 23 is given in (120), accompanied with some comments by Kallmeyer and Osswald 2013.

> Every base-labeled feature structure can be characterized by a finite conjunction of primitive labeled attribute-value descriptions. For instance, the frame structure of Figure 6 is specified by the following conjunction:

(120) $\boxed{0}$: causation ∧ $\boxed{0}$ · CAUSE : activity ∧ $\boxed{0}$ · CAUSE ACTOR ≜ $\boxed{1}$ ∧ $\boxed{0}$ · EFFECT : change-of-state ∧ $\boxed{0}$ · EFFECT RESULT : broken-state ∧ $\boxed{0}$ · EFFECT RESULT PATIENT ≜ $\boxed{2}$

> Note that the attribute-value matrix shown in Fig. 23 can be regarded as a normal form of the attribute-value description in (120), with conjunction symbols left implicit. Kallmeyer and Osswald (2013, 284)

I reuse the general structure of Fig. 23 for representing causative structures in the current project. The frame structure I use is shown in Fig. 24. It inherits from Fig. 23 the root event of *causation* having two subevents: a *cause* and an *effect.* According to RRG, the cause is the event of type *activity* having one participant, the effector. The effect subframe corresponds to the transitive verb, from which the causative is derived. In the most general representation, it is the event of any type (hence the *event* heading) having an *agent* and a *patient*. The linking of these frames to real language data is shown in Section 10.4.3.

8.3 TuLiPa: a parser for TWGs

To apply the described formalization to real data, one needs specialized tools, namely, a parser for syntactic analysis. Such a parser must support the newly introduced parsing operations (presented in Section 8.2.2) and also be compatible with frame semantics (as presented in Section 8.2.3).

Thus, a parsing environment that fully conforms to the formalized RRG described above required two main advancements: a formalism for handling TAG-like trees and an environment for processing frame semantics. The parser for RRG has been presented in Arps, Bladier, and Kallmeyer (2019) and implemented as an extension of the TuLiPA parser. Initially developed in Tübingen (hence the name: Tübingen Linguistic Parsing Architecture, TuLiPA), this parser was designed as a multi-formalism environment for various tree-based grammars. The paper by Kallmeyer, Lichte, Maier, et al. (2008) presents multiple of those, including one of the TAG versions (Gardent and Kallmeyer 2003). Once RRG has been formalized as a tree-wrapping grammar, it has been added to TuLiPa.

The algorithm suggested in Arps, Bladier, and Kallmeyer (2019) works bottom-up: it scans the lexical items, then builds layered RRG trees, starting with the lowest (closest to the lexical items) layers, and progressing to the root node. The root node's category is called *axiom* and must be provided in the parsing command. The parser applies operations to combine trees and the feature structures of nodes. The parser tries to compile all possible tree representations that are not discarded in case of failed unification.

As for the frame semantics, the tools for handling it have been developed separately. Indeed, pairings of syntactic and semantic formalisms can be very different. Namely, in Arps and Petitjean (2018), a solution for combining frame semantics with LTAG has been proposed. The main innovation of this solution was the usage of feature structures

$$\boxed{A0}\begin{bmatrix} \textit{causation} & & \\ \text{cause} & \boxed{E0} & \begin{bmatrix} \textit{activity} & \\ \text{effector} & \boxed{F0} \end{bmatrix} \\ \text{effect} & \boxed{B0} & \begin{bmatrix} \textit{event} & \\ \text{patient} & \boxed{C0} \\ \text{agent} & \boxed{D0} \end{bmatrix} \end{bmatrix}$$

Fig. 24. The frame used for causatives-of-transitive

in the semantic dimension instead of the previously used predicates.

8.4 Metagrammar: describing language descriptions

As shown in the previous section, the formalized grammar helps implement the theoretical findings. Namely, instead of manually describing every sentence, one can split them into fragments (elementary trees) and combine them with the help of parsing operations. The TWG formalism provides three operations that suit different syntactic situations. However, from the point of view of a tree-based grammar, the list of TWG elementary trees is still redundant and can be described more economically. For instance, all trees having four branches can be described as a single template, disregarding their linguistic content. Thus, this single template would describe lexical ditransitive constructions, three-argument causatives, and much more. This approach to description – describing not the language itself, but a set of descriptions – is called *metagrammatical*.

Currently, the term *metagrammar* is commonly used to denote a system that generates implemented grammars (written in any formalism). The term was first introduced in Candito (1996).

The key idea of a metagrammar is to express linguistic generalizations in order to factorize the redundant parts of the grammar (see discussion in Clément and Kinyon (2003)). Hence, the main gain from creating metagrammars is twofold. First, creating metagrammars reduces the time required to develop grammars for individual languages. The whole LinGO Grammar Matrix project (Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010) is centered on this idea: it provides precompiled libraries with cross-linguistic generalizations and a questionnaire to select the necessary options for the new target language. As a result, one obtains the so-called "starter grammar" (i.e., a first draft of a grammar) once the filled questionnaire is processed by the libraries.

The level of the metagrammar has been proven helpful for the development and merging of (comparatively) large-scale grammars of individual or closely related languages. Such investigations have been made by Fokkens, Avgustinova, and Zhang (2012) concerning CLIMB (Comparative Libraries of Implementations with Matrix Basis). According to the paper, "Germanic CLIMB (Fokkens 2011) was developed twice as fast as the comparable resource Cheetah (Cramer and Zhang 2009)" (Fokkens, Avgustinova, and Zhang 2012, 1678). The authors also remain optimistic about the time required to translate the grammars of other languages (namely, Russian and Mandarin Chinese) into CLIMB.

The second asset of using metagrammars to encode language descriptions is especially profitable for developing new linguistic theories (or applying linguistic theories to new data). Since metagrammars do not describe sentences themselves, but rather the sentence *descriptions* already, they can serve as a playground for testing several approaches to analyzing the data. This use case has received special attention in Fokkens (2011). The author presents a workflow for analyzing auxiliaries in German in Dutch. Within HPSG, they can be regarded as two distinct structures (Fokkens 2011, 1070–1071). The point is that both these ways of analysis can be encoded in a metagrammar and applied to the evaluation data. Hence, statistical results can prove that one of the analyses performs better (the other can thus be easily deleted from the final version).

To fulfill this second academic need, the language of metagrammatical description must be well-suited to it. Namely, it needs to offer a clear separation between the linguistic reality and the analysis, and a degree of modularity that permits the description of fragments of language structures. Early metagrammar languages and environments (Candito 1999; Xia, Palmer, and Vijay-Shanker 2005, see a detailed comparison of several metagrammatical systems in Crabbé et al. (2013, 622–624)) lacked these crucial properties. For tree-based grammars, such an environment never existed before XMG, the eXtensible MetaGrammar, suggested by Crabbé et al. (2013). (The LinGO Grammar Matrix is built upon HPSG and Minimal Recursion Semantics (Copestake et al. 2005) without using any tree representations.)

8.5 XMG: describing tree-based grammars with constraints

XMG is a language that allows one to describe trees, like those shown in Fig. 25. The paper by Crabbé et al. (2013) is usually cited as the reference source for XMG. It presents three main features of XMG as a language, see below. This paper uses FB-LTAG (Feature-Based Lexicalized Tree Adjoining Grammar) to illustrate its findings; however, many of these findings are also relevant for RRG, formalized as TWG.

> First, XMG is a declarative language. In other words, grammaticality is defined in an order-independent fashion by a set of well-formedness constraints rather than by procedures. In particular, XMG permits a fully declarative treatment of diathesis that markedly departs from the procedural rules (called meta-rules or lexical rules) previously used to specify tree-based grammars.
>
> Second, XMG is notationally expressive. The XMG language supports full disjunction and conjunction of grammatical units, a modular treatment of multiple linguistic dimensions, multiple inheritance of units, and a sophisticated treatment of identifiers. We illustrate XMG's notational expressivity by showing (i) how it facilitates the modeling of the tree fragment hierarchies required to specify tree-based grammars and (ii) how it permits a

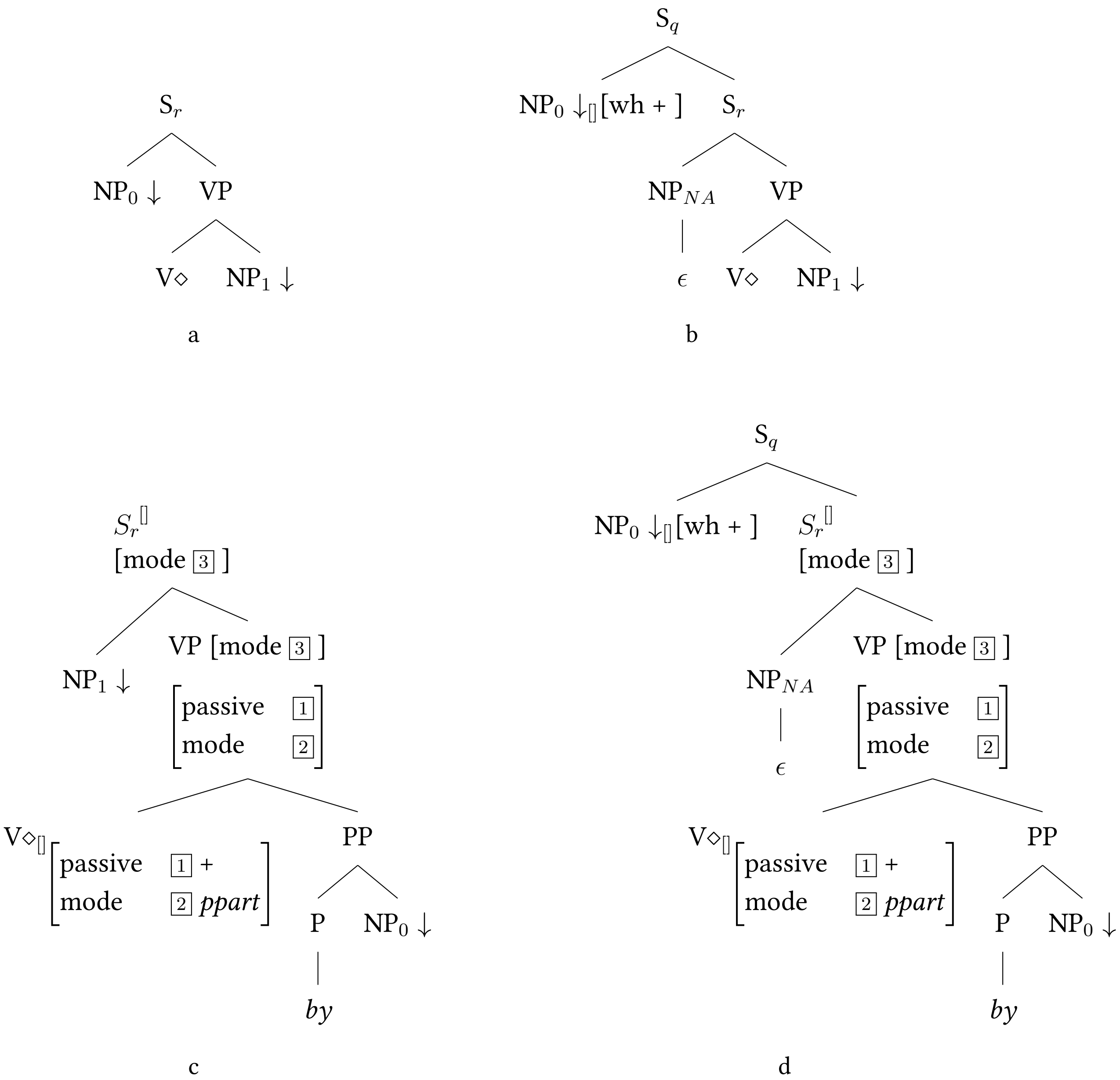


Fig. 25. Examples of trees generated by the classical XMG (reproduced from Crabbé et al. 2013, p. 595, Fig. 2)

> natural modeling of the syntax/semantics interface between semantic representations and syntactic trees as can be used in FB-LTAG.
>
> Third, XMG is extensible in that its computational architecture facilitates (i) the integration of an arbitrary number of linguistic dimensions (syntax, semantics, etc.), (ii) the modeling of different grammar formalisms (FB-LTAG, MC-TAG[31], IG[32]), and (iii) the specification of general linguistic principles (e.g., clitic ordering in French). Crabbé et al. (2013, 593)

The first feature – declarativity – is opposed to previous procedural approaches. This innovation holds for RRG as well. Instead of using commands such as *add*, *delete*, *move*, etc., XMG contains static instructions about what a tree looks like. The paper (Crabbé et al. 2013) illustrates this property of the language through its approach to verbal diatheses, namely, the passive (Crabbé et al. 2013, 597–599).

(121) a. Subject $\rightarrow$ CanonicalSubject $\vee$ Wh-NP-Subject

b. ActiveTransitiveVerb $\rightarrow$ Subject $\wedge$ ActiveVerb $\wedge$ CanonicalObject

c. PassiveTransitiveVerb $\rightarrow$ Subject $\wedge$ PassiveVerb $\wedge$ CanonicalByObject

d. TransitiveVerb $\rightarrow$ ActiveTransitiveVerb $\vee$ PassiveTransitiveVerb

To account for both active and passive diatheses of the verb, one must first define all the unique tree fragments that are used in both constructions. In (121), these are `Canonical Subject`, `Wh-NP-Subject`, `Active Verb`, `Canonical Object`, `Passive Verb`, `Canonical By Object`. Then, the necessary fragments are combined using a conjunction. Namely, in both trees for active and passive transitive verbs, the same notion of subject is used. What distinguishes `Active Transitive Verb` from `Passive Transitive Verb` is the different fragments used for verbs and objects. Finally, a family of trees (a set of trees in which the arguments of transitive verbs are realized in all possible ways) can be defined using disjunction (line (d) in (121)). The trees described by (121) are shown in Fig. 25.

This use case is especially relevant to my research since I also work with verbal derivations. However, this example is somewhat simplistic (as illustrative examples usually are), whereas my classes on real-world data are more complex.

The second important feature of the XMG language is notational expressivity. It means that XMG is capable of encoding language data in several *dimensions*, allowing coreferences not only within each dimension but also between them. The version of XMG presented in Crabbé et al. (2013) contains three dimensions: `syn` for describing syntactic tree-based grammars, `sem` for encoding semantics in the shape of flat semantic formulae (Gardent and Kallmeyer 2003), and `dyn` standing for a dynamic interface to al-

[31]Multi-Component Tree-Adjoining Grammar (Weir 1988)

[32]Interaction Grammar (Perrier 2000)

low unification of nodes and features belonging to different dimensions. XMG's innovation compared to earlier metagrammar engineering frameworks is its control language, which enables declarativity and notational expressivity. Namely, it supports exporting some variables, separating local usages from global ones. It means that by default, variables are local, i.e., accessible only within the class where they are defined. However, when needed, they can be exported and thus become visible to other classes. This approach solves the common problem of identifier sharing and renders the requirement for unique identifiers for all variables unnecessary. Technically, there are several ways to share information across XMG classes; they are presented in more detail in Chapter 9.

In the version of XMG I am using, the dimensions are different. Their addition was made possible by XMG's extensibility as a language and by the environment for developing new dimensions introduced in XMG-2 by Petitjean, Duchier, and Parmentier (2016). The initial architecture of XMG (Crabbé et al. 2013, 604–607) is built in such a way that a specific solver interprets each dimension. The way XMG descriptions are processed allows an arbitrary number of dimensions, enabling any desired combinations. Thanks to XMG-2, the scope of XMG-based projects has widened, both in terms of languages and the language phenomena covered.

Two extensions became especially relevant for my work. Initially, XMG suggested a `sem` dimension for encoding flat semantic structures. Several projects (for instance, Zinova and Kallmeyer (2012) and Kallmeyer and Osswald (2012)), however, preferred using frame semantics with tree-based syntactic representations. To accommodate these needs, a new dimension, `frame`, and a compiler, `synframe`, have been developed by Lichte and Petitjean (2015). Frames, by nature of their data structure, combine well with typed feature structures and the operation of feature unification. As a result, the new `synframe` toolkit (the dimension and the respective compiler) allows full control over variable handling across dimensions, without necessarily requiring any interface dimension. Second, in my solution, I extensively use two modules for describing lexical entries: `morpho` and `lemma`. I present them in more detail and with an immediate connection to my solution in Section 10.3.

To conclude the description of the XMG projects, it is important to note that work on aspects relevant to my research has been and continues to be carried out. The argument-linking has been approached by Kallmeyer, Lichte, Oswald, et al. (2016) – but only in the LTAG (not RRG) perspective and for some prototypical cases. The semantics of causation was modeled by Seyffarth (2019) – but only for English and only for one type of constructions (like *Sylvia laughed Mary off the stage*). The passive verbal derivation is present in all TAG-XMG projects, but other diatheses have not received as much attention (and TAG is still not RRG).

Non-European typologically different languages have received some attention among XMG developers. Namely, there is a TAG grammar of Arabic (ArabTAG Khelil, Duchier, et al. (2016) and Khelil, Zribi, et al. (2018)) – but it does not offer any comparison of languages. The same is true for the project modeling the verbal morphology of Ikota (Bantu, B25 in Maho (2009)) presented in Duchier et al. (2012) and Magnana Ekoukou (2014). The lexical derivation has been modeled by Andreou and Petitjean (2017) (the English *-al* suffix) and by Zinova (2017) (Russian verbal prefixes, see more in Zinova (2021)).

Thus, there has never been an XMG project aimed at modeling the construction of a verbal derivation across multiple typologically diverse languages using RRG analyses. For this reason, I am presenting every part of my solution in great detail in Chapter 10. However, before doing so, I suggest familiarizing with some most important concepts in XMG in Chapter 9.

8.6 Summary: theoretical and technical setup

The goal of my solution is to present output representations like the one shown in Fig. 26. The representation of each sentence comprises two parts: the syntactic tree and the semantic frame. These two parts are linked to each other, as shown by the labels. Consider, for instance, the label $\boxed{C0}$ corresponding to the patient of the event of killing, namely, *slug* in the semantic frame, and the same label assigned to the `NROOT` node linked to the morpheme *etana* 'slug'. The same label is also assigned to the nodes that govern the `NROOT` up to the `NP` to show that this morpheme denotes a participant of a verb that can act as its syntactic argument. All syntactic nodes are assigned feature structures that indicate their categories, morphological coding, and semantics. These features will be presented in detail in Chapter 10.

To produce such representations, several modules are needed. The greatest part of the syntactic and semantic analyses is written in XMG language within the dimensions `<syn>` and `<frame>`, respectively. These two dimensions can be compiled together by the `synframe` compiler; hence, it is possible (and very practical) to write them within a single file.

An important part of the `synframe` file is the metagrammar preamble. In this part, features are declared and given a type. Typing is essential for both syntactic and semantic dimensions to ensure the correct linking of constituents and error-free feature structures. The preamble can be part of the `synframe` or written in a separate `.mg` file. (In the latter case, one would need to use the special command `include` to make the preamble visible to the main `synframe` file.)

The `synframe` compiler takes the metagrammar file and produces all the models described by all *valuated* classes (more details on that are given below). These models usually correspond to elementary TWG trees. The output files of XMG compilers have the `.xml` extension. To combine these tree fragments into a single tree corresponding to the whole sentence, the **TuLiPa parser** takes the `.xml` file and applies parsing operations (Kallmeyer and Osswald 2023) to the models.

The parser does not compile abstract models from elementary trees. The goal of the parser is to analyze a sentence, which is fed to it along with the `.xml` files. The elementary trees do not contain lexical items, but instead have lexical anchors compatible with a set of lexical items (see Section 10.4.1 for details). This is another level of factorization of the grammar. Hence, the TuLiPa parser needs access to lexical files to provide a representation tailored to each sentence. Lexical files are created in `<morpho>` and `<lemma>` dimensions of XMG and are compiled by respective (different!) compilers. The output of these compilers in the `.xml` format is also fed to the TuLiPa parser.

Fig. 27 summarizes the theories and the tools I used to create my implementation and traces their relationship with each other and with the contributions I make in the present thesis. The classical RRG inspires the theoretical findings I presented in Chapter 7. Together with the TAG (Tree-Adjoining Grammar) formalism, it influenced Osswald and Kallmeyer (2018) (see above), which led to the formalization of RRG as a Tree-Wrapping Grammar (TWG). In addition, the TAG formalism influenced the creation of the XMG language, which is used by XMG compilers. The contribution I am presenting here is the set of metagrammars used to parse my target sentences. This contribution builds on my theoretical findings regarding the opportunities and limitations of the TWG formalism and the XMG language. All the contributions reflect real language data. The metagrammar files I wrote are fed to the XMG compilers, which generate fragments of analysis as XML models. Finally, the XML models and the target sentences are fed to the TuLiPa parser to obtain the final intended result, i.e., complete linked syntactic + semantic analyses of the target sentences in all studied languages (like in Fig. 26).

The remainder of the part is organized as follows. Chapter 9 is perhaps the most useful for understanding my implementation, since it outlines the key concepts of XMG, the language of my solution. Chapter 10 illustrates my solution by describing each module of it and explaining why each decision has been made. Even though the chapter is richly decorated with code snippets (see all of them in the List of Snippets), the full code is made available on GitHub and in Appendix A. Chapter 11 presents the evaluation of my solution and compares it to some similar projects. Some useful comments about the repository and instructions for reproducing my results are provided in Section 11.2. The last Chapter summarizes and discusses my findings.

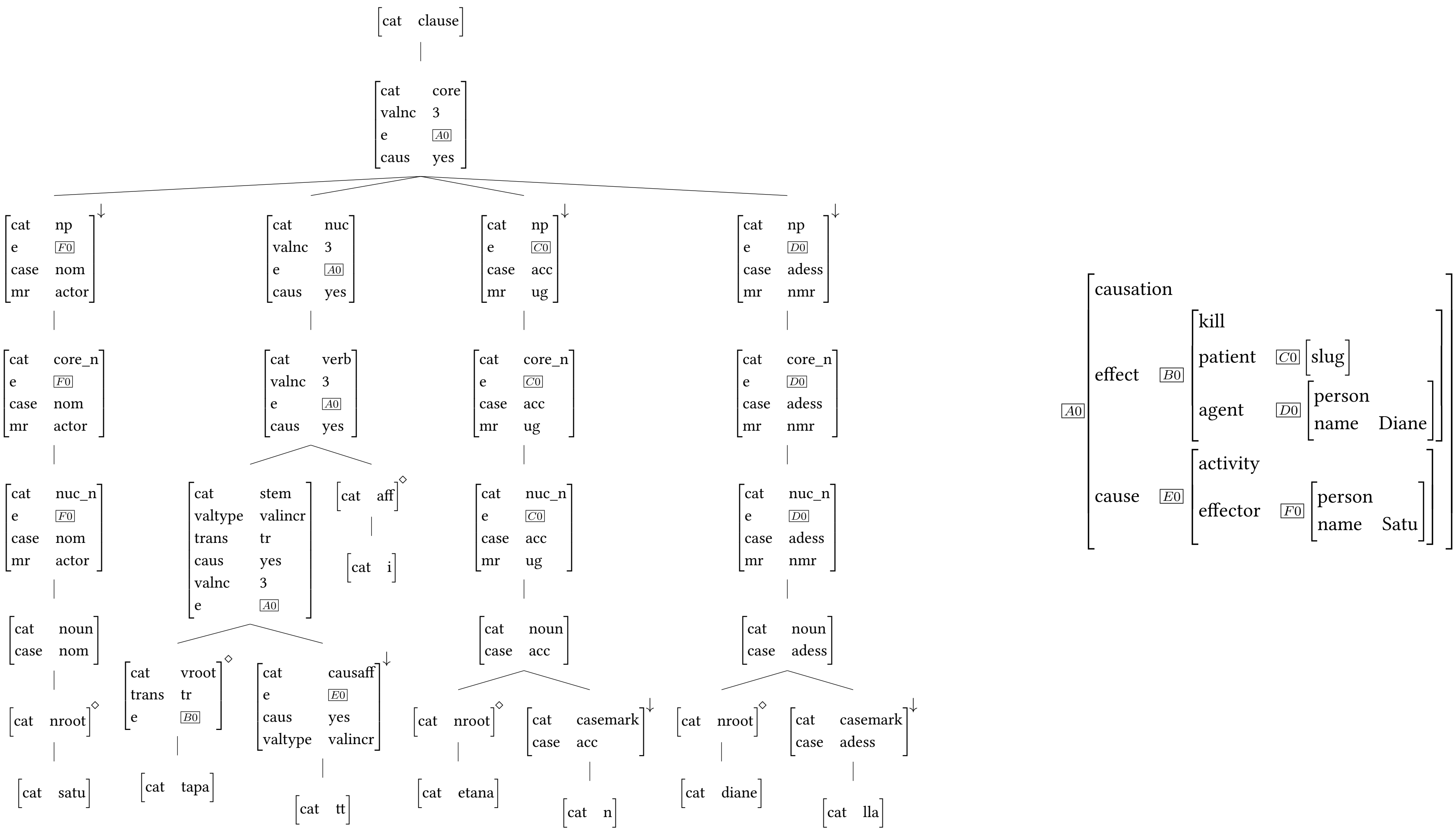


Fig. 26. The output representation for the Finnish sentence ‘Satu made Diane kill the slug.’

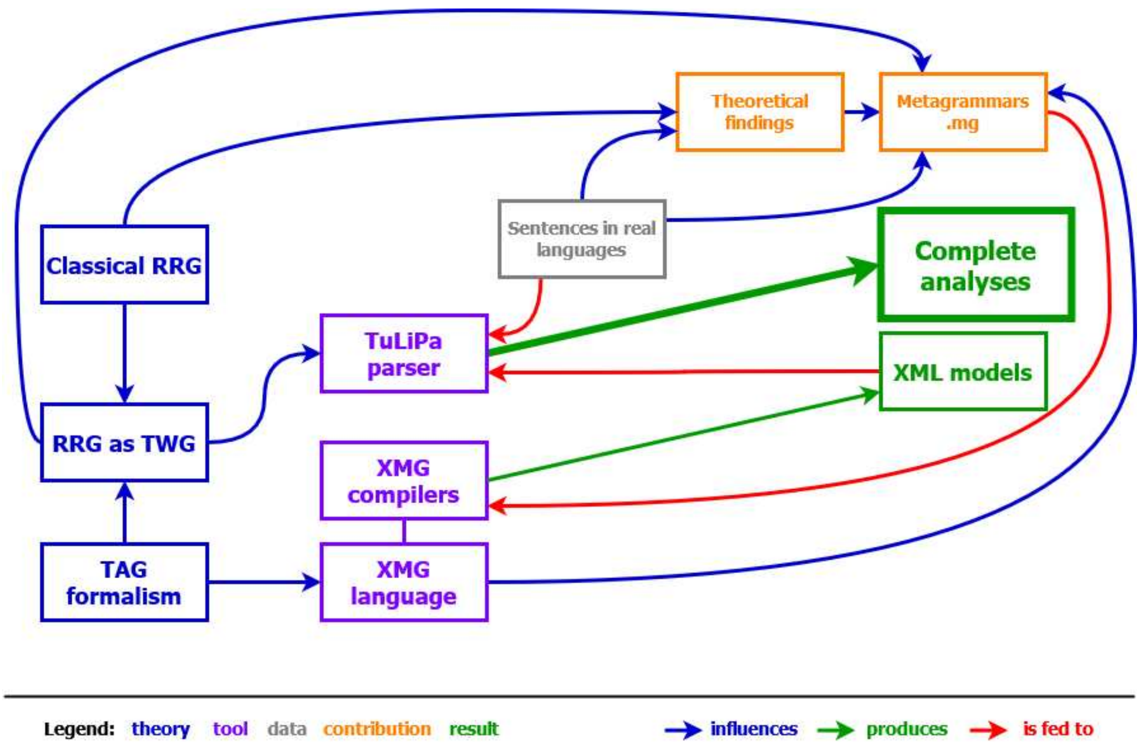


Fig. 27. The setup for the suggested solution

Chapter 9

Key concepts of XMG

The basic unit of the metagrammar's code is the *class*. One class describes a tree or a tree fragment, or imposes restrictions on a tree fragment. The general idea of the solution I provide is to make the metagrammar as modular as possible but nevertheless linguistically salient. This requires creating separate classes for distinct yet meaningful parts of my analysis.

As already mentioned, different dimensions at different levels of linguistic analysis require different XMG files, which are fed to different compilers. The next section illustrates and discusses the internal structure of a class for the `synframe` compiler. Classes for other compilers – namely `mph` (for morphology) and `lex` (for lemmas) – have a much simpler structure and hence are explained throughout Chapter 10. Basic XMG structures, such as features and variables, are common across all dimensions.

9.1 The internal structure of the class

XMG `synframe` classes have a complex yet straightforward internal structure, which is shown in Fig. 28. Each class opens with a class name statement containing the special command `class` and a freely defined name. The contents of a class (not including the commands, see below) are enclosed in braces.

A class can contain up to three other special commands that have to be given in the order shown in Fig. 28: `import`, `export`, and `declare`.

9.1.1 Import

The `import` command is used to add the contents of another class to a given class. If no additional class is required, one can simply miss this line.

The import mechanism takes all the statements of one class and adds them into another class as if they were written one after another. The class that is imported into

```
class NounWithCase
import NominalSpine[]
export ?NodeNoun ?Case ?LanguageFeatures
declare ?NodeNRoot ?NodeCaseMarker ?Case ?LanguageFeatures
{ <syn> {
    node ?NodeNRoot (mark = anchor) [cat = nroot] ;
    ?NodeNoun -> ?NodeNRoot ;
    {
        { node ?NodeCaseMarker (mark = subst) [cat = casemark , case = ?Case] ;
        ?NodeNoun -> ?NodeCaseMarker ;
        ?NodeNRoot >> ?NodeCaseMarker }
    |
        { ?Case = ?LanguageFeatures.UnmarkedCases }
    }
}   }
```

Legend: Class names Variables Properties Attributes Values Operators Commands

Fig. 28. The internal structure of an XMG class

another is often called the *daughter* class, the class to which another class is imported is sometimes referred to as the *mother* class.

Imported classes are referenced by names and followed by empty square brackets '`[]`'. Any number of classes can follow the `import` command, each followed by a pair of square brackets in turn. All of the statements of all the classes will be conjoint then. There is no possibility to use disjunction in the `import` part, i.e., one cannot tell a mother class to import *one or another* daughter class.

If one wants to import only some variables from a class, their names can be specified in brackets after the import command. However, this option is not used in my project.

9.1.2 Export

The `export` command tells what variables of a given class become accessible from the outside of it. The `export` command is followed by whitespace-separated variable names. Each variable name has to start with the '`?`' sign.

Usually, one exports the highest tree node from a syntactic dimension and the outermost frame from the frame dimension. Referencing this node, one could attach the given tree fragment to other trees (or tree fragments). The internal architecture of this tree remains inaccessible to other classes and thus cannot be modified even accidentally. Moreover, the local names do not get confused with the global ones, which makes it easier for human grammar developers to work with the variables.

The purpose of the `export` mechanism is thus to make specific variables – chosen by the grammar developers – accessible. This "on demand" approach offers the flexibility to use variables locally or globally, depending on their function in a specific project, without the limitations imposed by the grammar formalism's design.

In the example class in Fig. 28, one exports the variable ?NodeNoun, which happens to be the highest node in the described tree. The other two exported entities are not syntactic nodes but free-standing variables. They are exported to ensure the unification of the described tree fragment with other models. Technically, it is possible to use only local variables and never use the export command, but, in practice, something is exported from most classes.

9.1.3 Declare

The declare command is the least optional among these three. Each new variable (see details in Section 9.1.5) that appears in a given class must be declared, i.e., its name must follow the declare command. If a variable is used in a class but not declared, the XMG compiler throws an error. Variables that are imported from other classes, in contrast, do not have to be declared in this line.

In the example class shown in Fig. 28, there are four variables declared. Only two of them are exported; the others are reserved for internal use only. Conceptually, it protects the structure of the ?NodeNoun from external modification. Note that the variable ?NodeNoun is not declared. The reason for that is that this variable has been **exported** from the class NominalSpine, then **imported** into the class NounWithCase and **accessed** in the respective line.

9.1.4 Dimensions

After these commands, the body of the class starts. It is enclosed in braces '{ }'. The body of the class usually contains one or more dimensions. The names of the dimensions are written in angle brackets, see <syn> in Fig. 28. Each dimension opens its own block of instructions, also enclosed in braces. Although dimensions have different notation requirements, some variables can appear outside any dimension.

The example class shown in Fig. 28 has only the syntactic <syn> dimension. In my metagrammar, I also use the <frame> dimension to describe semantic structures (see details on frames in general in Section 8.2.3 and on the development of XMG towards an architecture with a single synframe compiler in Section 8.5).

Many remarks concerning the XMG language and its operations hold for both these dimensions. Namely, one can have variables and features in the frame dimension, conjunction and disjunction are supported, and variables can be imported and exported. Here, I would like to make some notational remarks to facilitate the reading of the frames used later in the present work.

```
1  class example
2  declare ?X1 ?X2
3  {
4    <frame>{
5      [causation,
6       actor:?X1,
7       theme:?X2,
8       cause: [activity,
9               actor: ?X1,
10              theme: ?X2],
11      effect: [activity,
12                    actor: ?X2]
13      ]
14    }
15 }
```

Snippet 1: An example class illustratung the frame dimension (taken from the XMG documentation)

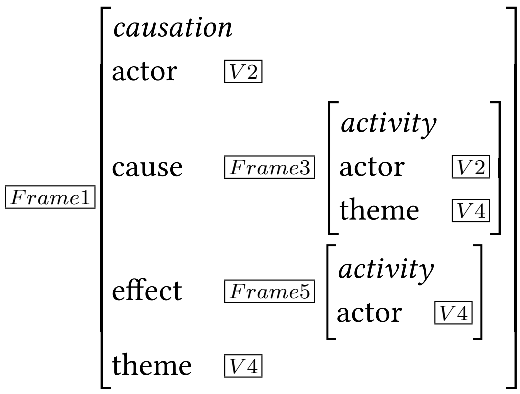


Fig. 29. The model generated by Snippet 1

Consider Snippet 1 (taken from the XMG documentation[33]). One can note that the dimension-external part of the class remains the same: there is a class name and a `declare` command followed by a pair of curly brackets enclosing the body of the class.

Each frame within the `frame` dimension is enclosed in a pair of square brackets. As shown in Fig. 29, the graphical output of Snippet 1 shows that frames can be nested, i.e., some frames can appear inside other frames.

The first line of each frame specifies its type (see *causation* in line 5, *activity* in lines 8 and 11 in Snippet 1). These types have to be listed in the metagrammar preamble as `frame-types`. The remainder of each frame is organized as attribute-value pairs. Attributes are given first, and values follow after the colon ':' sign. As shown in Snippet 1, values can be either variables (declared in the same class) or other frames.

The operation of conjunction within the frame dimension is indicated with commas. Disjunction is not illustrated in Snippet 1, but is supported for frames and functions the same way as in the `<syn>` dimension.

[33] `https://github.com/spetitjean/XMG-2/wiki`, accessed on 01.05.2026

Projects focused on advanced semantic modeling also use `frame-constraints` and `frame-relations`. Frame constraints help organize frames into hierarchies, providing information about subtyping, (in)compatibility, and attribute requirements. Frame relations declare "relations between unification variables" (see XMG Documentation). In my work, I do not create any frame hierarchy. Therefore, the usage of these instruments is not required for my project. Nevertheless, to show that my solution can be expanded to handle more complex systems of frames, I introduced one frame constraint (`frame-constraints = cause -> effector:+`) telling that any frame of type `cause` must have an attribute `effector`. More constraints can be added to this field as the metagrammar requires.

In my metagrammar, I also use two other dimensions: `<morpho>` and `<lemma>`. They still use basic notions of the XMG language; however, their internal structure and notational conventions differ from the dimensions described above. Therefore, I prefer not to overload the present section with these details. The relevant notions are introduced in the respective sections together with the description of my solution (Section 10.3.1 and Section 10.3.2 respectively).

9.1.5 Variables

The notion of *variable* has already been mentioned several times. A variable can function in different ways within a class. If a variable is supposed to function as a tree node, it has to be preceded by the `node` command. (More details on nodes are given below in Section 9.1.6.)

Variables can also function as undetermined values of attributes. Consider the statement `[case = ?Case]` in Fig. 28. It tells that the node `?NodeCaseMarker` has the feature `case` assigned, but its exact value is not yet known. Later, this variable gets unified with an explicit feature value.

Free-standing variables sometimes function as attributes. These variables can take either explicit values (e.g., `?Case = nom`) or other variables as undetermined values. One more option to assign a value to an attribute is to reference a value of a feature assigned to a different variable via the *dot notation*. This is exemplified in Fig. 28: `?Case = ?LanguageFeatures.UnmarkedCases`. This statement tells that the free variable `?Case` takes the same value as the feature `UnmakedCases` assigned to the variable `?LanguageFeatures` in some other class. The value remains underspecified until the class shown in Fig. 28 gets somehow unified with a class containing the variable `?Language Features` with the specified value of the attribute `Unmarked Cases`[34]. The mechanism of the dot notation is explained in more detail in Section 9.5.

[34]In the metagrammar, names of variables, classes, and features must not contain whitespaces. In the text of this chapter, I introduce whitespaces for better readability.

```
feature cat : CAT
feature mr : MR
feature case : CASE

feature caus : BOOL
feature peri : BOOL

feature i : LABEL
feature e : LABEL
```

Snippet 2: Excerpts from the preamble illustrating different kinds of features

9.1.6 Nodes and edges

Nodes are entities of a special type `node` that can function as constituents of the syntactic tree. Nodes can be assigned feature structures, such as variables. Unlike free-standing variables, a node can also have properties, consider `(mark = anchor)` in Fig. 28 (see more in Section 9.3).

Some special operators encode graph edges, i.e., relations between nodes. If the relations between nodes are underspecified within the class, the compiler generates as many models as possible without violating any constraints. In the current project, only two operators are used (more types are described in the XMG documentation)[35]. The statement with an operator of strict dominance '`->`' tells that the first node in the statement is the direct parent of the second node in the statement. In Fig. 28, the node `?NodeNoun` has two children nodes: `?NodeNRoot` and `?NodeCaseMarker`. The second operator, '`>>`', encodes direct precedence. It specifies the linear order of nodes at the same tree level. Thus, in Fig. 28, it orders the two children of the `?NodeNoun`: `?NodeNRoot` precedes `?NodeCaseMarker`.

9.1.7 Closing remarks

Indentation plays no special role in XMG. Brackets can be placed on any line and with any number of indents as long as they open and close correctly.

If one needs to make the output of a class visible to the parser, one must *valuate* it: insert a command `value` followed by the class name. Usually, all valuation clauses are written at the end of the main `.mg` file. One can also write them in a separate file and `include` elsewhere.

9.2 Features

In XMG, a feature is a typed attribute-value pair. Features are defined in the preamble and then are assigned to variables. Features appear in all dimensions of XMG, and their

[35] `https://github.com/spetitjean/XMG-2/wiki`, accessed on 01.05.2026

unification determines the structure of complex models[36].

Some examples of feature declaration are given in Snippet 2. The types of the first three features are manually defined elsewhere in the preamble. The second group demonstrates boolean features, a type that is already available in XMG. The last two features are of type `LABEL`, they are responsible for relating components of representations to each other and producing the labels like $\boxed{C0}$ in Fig. 26. Some technical features do not need to be declared in the preamble because they are implicitly defined by the language. These are features used in lexical dimensions `morpho` and `lemma`, see more details in Section 10.3.

Features can be assigned to class entities, nodes, or free variables. A variable can have any number of features assigned, but each attribute must be used only once, with a single value. Thus, a node or a variable is determined by its feature structure. In this respect, each node or variable receives a type corresponding to all the feature types it has been assigned. For example, the feature structure `[cat = v]` is of type `[cat - CAT]`, and the feature structure `[cat = v, caus = yes]` is of type `[cat - CAT, caus - BOOL]`. Each of these types is an instance of a general type *feature structure*.

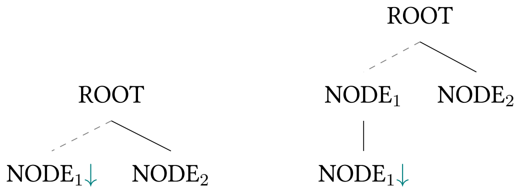
The essential mechanism behind features is *unification*. The idea of the unification is that all attributes tend to obtain the same value when possible, resulting in the smallest (hence, computationally optimal) model. Tab. 26 shows some examples of feature unification. When there are two variables or nodes with exactly the same feature structure (like in the first line of Tab. 26), once they occur in the same class (e.g., their classes are imported together), they produce a single node with the same feature structure. The same happens if one of the values is given as a variable (as in `ClassB` in the second line in Tab. 26). The undetermined value provided as a variable is replaced by the explicitly assigned value from the other class. In the situation when feature structures of the two nodes are entirely (as in the third line in Tab. 26) or partially (as in the fourth line in Tab. 26) different, no unification takes place. Thus, in the class importing both `ClassA` and `ClassB` in lines 3 and 4 of Tab. 26, both nodes `?A` and `?B` are kept.

ClassA	**ClassB**	**import ClassA[] ClassB[]**
node ?A [case = acc]	node ?B [case = acc]	node ?A [case = acc]
node ?A [case = acc]	node ?B [case = ?c]	node ?A [case = acc]
node ?A [case = acc]	node ?B [case = dat]	unification failed
node ?A [case = acc, mr = a]	node ?B [case = acc, mr = ug]	unification failed

Table 26: Feature unification

A special instance of feature unification is *percolation*. This effect is visible in trees, where the same attribute is assigned to different levels, and its value is not specified,

[36] I have experimented with assigning features of different kinds to various structural elements in different languages in Generalova (2020e). However, these experiments did not form a solid part of the current thesis, even though they influenced my line of thought.



a. Intended output b. Real output

```
{<syn> {node ?ROOT [cat = root];
        node ?Node1a (mark = ddaughter)
            [cat = node1];
        node ?Node1b (mark = subst)
            [cat = node1];
        node ?Node2 [cat = node2];
        ?ROOT -> ?Node1;
        ?Node1a -> ?Node1b;
        ?ROOT -> ?Node2;
        ?Node1b >> ?Node2   }    }
```

c. Sample code

Fig. 30. Technical limitations of properties

i.e., it is given as a variable. Once the determined value becomes available – usually at the lowest node of this tree – all the variables used as placeholder values are unified, i.e., receive the same attribute value. Hence, the determined value *percolates* from the bottom of the tree upwards.

9.3 Properties

A property is a special kind of feature assigned to a node. Property, like feature, is also a typed attribute-value pair declared in the preamble.

In my solution, I use one property `mark`. It plays no role within the metagrammar but gives instructions to the parser. Namely, it tells the parser whether a node is eligible for parsing operations. The values of this property and their graphical representations are listed in Tab. 27.

Code	Output	Function
`(mark = anchor)`	NOUN◇	This node can anchor a morpheme.
`(mark = subst)`	NOUN↓	A tree can attach to this node.
`(mark = star)`	NOUN*	This node (and the tree below it) can adjoin to another node with the same category.
`(mark = ddaughter)`	NOUN	This node can be separated from its tree while wrapping.

Table 27: Values of the property `mark`

There is a technical limitation that only one property of the same kind can be assigned to a node. Therefore, if the same node participates in more than one parsing operation, several nodes must be declared in the code, each with its own mark. This limitation is illustrated in Fig. 30. The first figure, Fig. 30a, shows the intended output. In this tree, one would like the node of category `Node1` to be marked both as a substitution node and as a d-daughter (daughter of the dashed edge indicating the possibility of wrapping). However, the same node cannot have two marks assigned to it. Therefore, one needs to declare two nodes of the same category that are placed one below the other. In the code Fig. 30c, one can see nodes named `?Node1a` and `?Node1b`, both having the category `node1`. The upper one, `?Node1a` has `(mark = ddaughter)` assigned, and the lower one receives `(mark = subst)`. The two nodes are connected with the usual dominance edge. They are assigned the same set of features, meaning the feature values can percolate from the lower node to the upper one. Therefore, this architecture does not cause any information loss. Although this architecture is slightly verbose, it is nevertheless the minimal possible given the technical requirements of the XMG language.

9.4 Logical operations

9.4.1 Conjunction

XMG supports two logical operations: conjunction and disjunction. In all dimensions, the semicolon sign '`;`' is placed between conjoint statements. On the one hand, it indicates the end of a previous statement and the beginning of the new one. Simultaneously, it tells that both statements hold, i.e., indicates their conjunction.

Within feature structures, attribute-value pairs are separated with a comma '`,`' having the same twofold semantics as the semicolon above.

Another way of conjoining statements is `import`: all the statements of the imported class are considered to be conjoint with all statements of the class to which it is imported.

9.4.2 Logical disjunction

Disjunction is supported in two varieties. The 'usual' logical disjunction is introduced with the '`|`' sign. It separates two statements, meaning that either one or another of them holds (i.e., the disjunction is exclusive). However, there are situations in which both statements seem to hold because one subsumes the other. For instance, both parts of the disjunction `[case = ?c | case = acc]` are true when `[case = acc]`.

In practice, it means that a class with a single disjunction of two clauses normally produces two output models. Subsequently, if more than two elements are disjoined or if there is more than one disjunction, the number of models is scaled accordingly.

Logical disjunction is used several times in the presented solution, see Section 10.3.1.1, Section 10.4.2.1, Section 10.4.3.1, and some other mentions.

9.4.3 Atomic disjunction

There is also a second type of disjunction, the so-called *atomic disjunction*. It is introduced by the "`@`" sign.

The notation `?Noun [case = @{nom, acc}]` tells that the feature `case` is assigned to the variable `?Noun` can unify with another feature `case` that has either

```
1   type CASE = {abl, abs, acc}

3   * case != acc

5   case = @{abs, abl}
```

Snippet 3: Atomic disjunction in the absence of negation

nom or acc as value.

Although it might seem to be just “syntactic sugar”, atomic disjunction works quite differently from the usual logical disjunction and thus is preferable for some situations to better reflect the phenomena attested in real-world languages. Instead of creating several models, the atomic disjunction keeps multiple values of a given attribute at the same time. It means that only one tree is produced, but it is not known which feature would appear at a given node. Thus, the logical disjunction in the feature structure `[case = acc | case = dat]` creates two models: `[case = acc]` and `[case = dat]`. The atomic disjunction `[case = @{acc, dat}]` creates one single model able to unify with another structure with either `[case = acc]` or `[case = dat]` (but not with, say, `[case = abs]`). If the structure with an atomic disjunction does not get unified with anything, it remains underspecified. In this respect, it is similar to the use of a free variable as a value, such as `[case = ?c]`, but is more constrained.

To sum up, one might consider atomic disjunction as a way to constrain feature unification, but not as a method for creating more models. In the current project, I use this type of disjunction to account for cases that can be unmarked in a given language, see details in Section 10.5.2.

9.4.4 Negation

The third operation of the classical Boolean logic is negation. As XMG describes models, negation is not expected to be widely used. Indeed, the statements that do *not* hold are irrelevant to the description. Moreover, negation complicates the constraint-based architecture, making it difficult to satisfy all the constraints and, thus, to compute models. However, sometimes negation could be a helpful shortcut for defining values.

For example, to address all cases except nom, one could wish to write something like `case != nom`. This kind of negation is nevertheless not supported either. Since all the features are typed, and the accepted values of attributes are listed in the preamble, when one needs to negate a given value, one should use an (atomic) disjunction of all the other values. For instance, if there are three cases listed in the preamble (see line 1 in Snippet 3), the synonym for the impossible line with negation (line 3) would be the disjunction of the two remaining values (line 5). Atomic disjunction suits these purposes better than logical disjunction because it creates one single model that can finally take the unique value without overcomplicating the output structure.

Another type of negation one might wish to implement is the negation of unification. It means that one might wish to explicitly prohibit the possible unification of two variables. Again, negation is not supported; therefore, one must describe two entities in such a way that they do not become available for unification. Usually, one assigns them the same feature with different values, which prevents unification. However, it is not always the case that features introduced for such a purpose reflect the language reality or the preferred analysis.

To sum up, some usages of negation can be reproduced using only conjunction and disjunction in XMG, but the support of a proper negation could nevertheless be helpful.

9.4.5 Implication

A more complex yet very useful logical operation is implication. It helps to describe several models that have something in common more efficiently. Implication is not supported in XMG. This reflects the indicative (in linguistic terms) purpose of the XMG language. XMG code is not a command to create something new; it describes what already exists. Thus, in a static description, there is no place for conditional statements.

Nevertheless, some parts of XMG descriptions depend on each other and thus are better accounted for as groups. One possible solution would be to describe each bit of information in a separate class and afterward import these fragments into larger classes. Although it is in full accordance with the modularity principle in XMG, such excessive fragmentation makes the code difficult to read. Creating such a code can be challenging because separating linguistically salient models into fragments that share common traits is not straightforward. These fragments could share several axes of variation, requiring an especially complex division of information across several units of code. Moreover, to function as independent code units, these fragments would require a well-developed system of export-import constraints, which can be difficult to maintain.

Given all these complications, one would prefer having the option of introducing several conditionally motivated variants of a model within a single class. This type of implication can be represented as a conjunction of disjunctions. In Snippet 4, an artificial example that illustrates the representation of the implication as a disjunction of conjunctions is shown. Lines 5 and 6 are parallel structures of two statements: the first claiming that the value of some boolean feature is positive, the second assigning a given value to a free-standing variable. In these lines, the first claim can be read as the premise, and the second as the consequence. (Logically, there is no ordering constraint here: one can read the second claim as the premise and the first one as the consequence.) Once put in words, lines 5-6 could look like this:

```
class CauseeCoding
...
{<syn> {
{
    {?CauseeIsAdjunct = no; ?CauseeCase = dat} |
    {?CauseeIsAdjunct = yes; ?CauseeCase = loc}
}  }  }
```

Snippet 4: Implication presented as disjunction of conjunctions

(122) a. If the causee is an adjunct, it receives the locative case.

b. If the causee is not an adjunct, it receives the dative case.

The code reflects that each pair of values is assigned to this pair of variables, and that other pairings are prohibited.

One can note that no *else*-clause can be formulated within this approach since there is no negation. To formulate an *else*-like condition, one would need to disjoin all the values that were not used in previous 'conditional' claims (which compromises the whole idea of the *else*-clause).

9.5 Dot notation

```
class ClassA
export ?A
declare ?A ?X
{ ?A [cat = noun, case = ?X] }

class ClassB
import ClassA[]
declare ?B
{   ?B [cat = noun];
    ?B.case = ?A.case
}
```

Snippet 5: Example of using the dot notation

There is an opportunity to reference feature structures and even individual attributes from other classes without importing them. This can be done by providing the path to the selected feature using the *dot "." notation*. Such precise addresses are needed to avoid potential conflicts, for example, when multiple features of the same kind exist within a class.

Consider Snippet 5. First, in `ClassA` there are two variables declared – `?A` and `?X` – but only one of them is exported. The variable `?X`, standing as a value for the attribute `case`, remains inaccessible to other classes. It can, nevertheless, be addressed using the dot notation, which is demonstrated in `ClassB`. It imports `ClassA`, but since only `?A`

```
1  class LanguagePlugin
2    export ?LanguageFeatures
3    declare ?LanguageFeatures
4    {?LanguageFeatures = [
5      BoolCauseeLikeRecip = yes,
6      BoolCauseeLikePassDemAg = no,
7      MorphPsaCase = erg,
8      MorphUgCase = abs,
9      MorphRecipCase = dat,
10     MorphPassDemAgCase = ins
11   ];
12   <iface>{[_lang_ = test]}}


15 class CausOfTrans
16 ...
17 { ...
18 <syn> { ?AffecteeCase = ?LanguageFeatures.MorphUgCase;
19 {
20     {?LanguageFeatures.BoolCauseeLikeRecip = yes;
21     ?CauseeCase = ?LanguageFeatures.MorphRecipCase} |

23     {?LanguageFeatures.BoolCauseeLikePassDemAg = yes;
24     ?CauseeCase = ?LanguageFeatures.MorphPassDemAgCase
25 }  }  }  }


28  class Intersection
29  import CausOfTrans[] LanguagePlugin[]
```

Snippet 6: Using the dot notation for accessing language-specific information

was exported, it is the only variable imported. The variable `?B` is declared in `ClassB` and assigned a feature structure (see line 9). In addition to the argument `cat`, it can be ascribed the argument `case`. If one would like it to receive the same value as the `case` attribute of `?A`, one can use dot notation to specify the path to the value by referring to the feature structure and the attribute to be accessed. This mechanism helps address the values of features having access to the variables to which they are ascribed.

After exploring a toy example, I would like to demonstrate the real use of the dot notation in my solution. Consider Snippet 6 comprising fragments of three classes. The first class, entitled `LanguagePlugin`, contains one single feature structure called `?Language Features` with many attributes. The values of all attributes are given explicitly and correspond to real values attested in a given language. The second class (lines 15-25) presents a fragment of a cross-linguistic description of causatives-of-transitives. Namely, it tells that in all languages, the affectee case is the same as the case of the prototypical undergoer. As for the causee, it can be assigned two different cases depending on the type of construction attested in a language. Note that the formulations in the class `CausOfTrans` correspond to generalizations drawn in Part II and reflect linguistically salient observations formulated with comparative concepts.

From the technical point of view, all the variables used in the class CausOfTrans are declared locally since nothing is imported. Hence, within the class CausOfTrans alone, the formulation ?LanguageFeatures.BoolCauseeLikeRec is no different from the formulation ?A. However, the naming becomes crucial when the two classes – LanguagePlugin and CausOfTrans – are imported together into a third class called Intersection (lines 28-29). Within this class, the variable ?Language Features from the LanguagePlugin becomes visible to the class CausOfTrans, and the addresses via dot notation end up unifying with the explicitly given values. Namely, the language described in the ?Language Features has only one of the two constructions described in CausOfTrans, hence only one of the two options listed in CausOfTrans is produced. In this model, the affectee will receive acc case, and the causee will receive dat case.

Thus, I consider dot notation a powerful instrument for tracing relations between various constructions without affecting them. It is also a powerful tool to capture cross-linguistic generalizations independently of language-specific information. To ensure this method works correctly, all necessary variables must be exported and thus externally visible. To access the value of an attribute, one must have the two classes in question imported (or otherwise conjoint within a single class).

Chapter 10

The metagrammar of three-argument constructions with a morphological causative

10.1 Overview

This chapter presents the implementation of my theoretical findings as a metagrammar[37]. The goal of this implementation is to produce linked syntactic and semantic RRG analyses of real-world sentences in different languages. This chapter gives all the details about how this implementation is built. First, it offers a high-level overview of the components, then explores questions of data processing, and finally presents all classes of the metagrammar with extensive comments on their purpose and structure.

To effectively describe cross-linguistic generalizations related to a specific grammatical phenomenon, and to ensure accurate parsing of target sentences in the languages being studied, the proposed solution must encompass three key components. First, it requires language-specific information for each target language. Second, it needs a comprehensive linguistic analysis of the phenomenon in question. Finally, the solution must include rules that allow for the application of these analyses across the various

[37] Early versions of the solution suggested in this chapter have been presented as conference talks (Generalova 2020a; Generalova 2020d). The prototype of the multilingual architecture comprising construction classes and language plugins has been demonstrated in Generalova and Petitjean (2020). The architecture idea is mine, and I am grateful to my co-author, Simon Petitjean, for helping me with the technical matters and writing the review section of the paper. The development of my ideas can be found in some proceedings papers about language plugins (Generalova 2020f; Generalova 2021a) and construction classes (Generalova 2021b). In Generalova (2021c), I have introduced the class `TreeShape`, which can be regarded as an ancestor of the class `All Range Causative Constructions` demonstrated in this thesis. The treatment of morphology as syntax received special attention in Generalova and Petitjean (2022). I am once again grateful to my co-author, Simon Petitjean, for being responsible for the technical accuracy of the paper, for writing the review and the comparison with competitors (Sections 3.2, 3.3, and 3.4), and for taking care of the illustrations. None of these sections or illustrations appears in my present thesis; I have created similar figures using my own skill and resources.

languages. In simpler terms, this means that at a minimum, one must provide a lexicon and grammar for each language, along with a “universal grammar” that covers causative constructions.

In XMG, the lexicon is encoded in two dimensions; hence, two lexical files are required for each language. More details on that are offered in Section 10.3.

The “grammar” (at least, the list of grammatical features relevant for describing causative constructions) of each language is stored separately. Since it has to be easily accessible for the syntactic part of the whole metagrammar in order to allow parsing sentences in accordance with linguistic generalizations, the decision was made to create for that a special type of class in the **syntactic** dimension. These classes were named *Language Plugins*. It is inspired by the fact that they store language-specific data that plugs into generalized linguistic analyses at the latest stage to enable parsing. See Section 10.5 for a detailed description of how they are organized and linked to other classes.

As stated earlier, each file type is fed to a separate XMG compiler. In the architecture I present, the number of lexical files (`<morpho>` dimension fed to `mph` compiler and `<lemma>` dimension fed to `lex` compiler) is equal to the number of languages involved. Example files of this kind can be found in Appendix B.

In addition, the architecture comprises one file fed to the `synframe` compiler, also referred to as *the main metagrammar* (see the full file in Appendix A). In addition to the already mentioned Language Plugins, it contains the *preamble* – a part of the metagrammar where all types and features are defined. Finally, it presents a modular implementation of my theoretical findings, presented class by class throughout this chapter.

To parse a particular sentence, one thus needs to feed TuLiPa four types of data: (1) the output of `mph`, (2) the output of `lex`, (3) the output of `synframe`, (4) the target sentence in the form of whitespace-separated morphemes. See more details of this setup in Section **??**.

10.2 Data preparation

The unit of my metagrammar is a morpheme. This decision is motivated by the fact that most of the languages in my sample have rich morphology. Hence, each morpheme makes its contribution not only to the meaning of the word, but also sometimes to the whole construction. Therefore, expanding the syntactic description to cover the morphological level is a smooth solution for representing this information.

In this line, each class of each lexical file in each language corresponds to one morpheme. (In some cases, I had to simplify the glossing of examples and pretend several morphemes to be single, see below.) All morphemes are introduced as strings of non-whitespace Unicode characters. Lemmas are given either as English words (for lexical morphemes), or as Leipzig glosses (for grammatical morphemes).

10.2.1 Sentence segmentation

Since the unit of my metagrammar is a morpheme, the input sentence must be preprocessed so that each morpheme is separated from others by whitespaces. Boundaries between words are thus no different from boundaries within words. The correct attachment of the morphemes to each other is due to their categories, which are also syntactic features (see Section 10.3.2.2 for details).

Let us explore some steps concerning morpheme segmentation and morpheme annotation using examples (123) and (124).

(123)	ł’əźə-m	ś’āla-m	pχa-r	∅-yə-ryə-ġa-q’ʷət-a-š’	KABARDIAN
	old.man-ERG	boy-ERG	tree-ABS	3SG.ABS-3SG.IO-3SG.A-CAUS-cut-PST-AFFMT△	

‘The old man made the boy cut the tree.’ Matasović (2010b, p. 50, (18))

(124)	ytyk	p‘-oɣla-aχ	pitɣy	ama-gu-d	NIVKH
	father	POSS-child-DAT/ACC	book	see-CAUS-FIN	

‘The father showed his son the book.’□

Nedyalkov, Otaina, and Xolodovič (1969, p. 192)

In (123), the morpheme *-r* ‘ABS’ is a case marker. It can attach only to nominal roots and must follow the root itself – this information is encoded in a metagrammar class describing this morpheme. There is no prefix *r-* in the XMG lexicon of Kabardian that would attach to the last word of the sentence. Therefore, the sequence “pχa r” in the input is interpreted correctly as ‘tree-ABS’ without any alternative.

The morpheme *p‘* ‘POSS’ in (124) is treated somewhat differently. Since nominal morphology unrelated to argument marking is outside of the scope of this metagrammar, all nominal categories are merged into a single one called *nominal inflection*. Morphemes of this category attach to nominal roots via sister adjunction and thus can occur in any position relative to the root (see details in Section 10.4.2.2). Because of this simplification, the morpheme sequence “ytyk p‘ oɣla aχ” can be read by the parser as two different lines: correctly as “ytyk p‘-oɣla-aχ” and with incorrect attachment of POSS to the preceding (not the following!) nominal root, “ytyk-p‘ oɣla-aχ”. This complication is not inherent to the architecture of the metagrammar but comes from an oversimplification of nominal morphology. In a real-life project accounting for all types of nominal morphology equally, the namesake restrictions on morpheme positioning would have applied systematically. Namely, one could have introduced special features such as `suffix` or `prefix` to systematically apply order constraints to all morphemes. In my current solution, I decided to continue over-simplifying and pretend this morpheme to be part of the nominal root: “p‘oɣla” ‘one’s child.’ This aspect does not affect the analysis of causative constructions and aligns with the present work’s focus, which deliberately avoids addressing less relevant topics.

As to some possible objections to the very fact of feeding the parser with the morpheme-segmented examples, I regard this decision as an economical yet practical compromise. Since the metagrammar is mainly intended as a resource for linguists, I found it perfectly appropriate to stick to the transcriptions and data formats used by linguists and found in linguistic papers. Therefore, I use glossed texts in Latin-based transcriptions. If one would like to parse sentences from real texts and/or unglossed corpora, they could use third-party solutions for the respective language to pre-process their data. There are solutions that automatically segment language data into morphemes. Some tools aimed at segmenting lesser-resource languages have been developed by SIL International[38] for field linguists (e.g., FLEx or Field Linguist's Toolbox). Similar solutions targeting corpora developers exist as well.

The question of whether XMG, as a framework, needs its own tool for segmenting words into morphemes is to be addressed once more attention is drawn to describing morphologically rich languages.

10.2.2 Conversion from glossed examples to `.mg` files

The preparation of lexical files for the metagrammar requires quite some work, given the special notation and structure of `morpho` and `lemma` XMG classes. This could have significantly increased the time for the metagrammar development. However, a large amount of the necessary annotation is already contained in the language itself and in non-XMG formats traditionally used for its representation, namely in interlinear glossing. Interlinear glossing is a well-known standard used in different branches of linguistics. Hence, its use as a source format for creating metagrammar files can bring together very different projects and enrich the metagrammar with more data. Thanks to the information encoded in interlinear glosses, files of the metagrammar can be automatically generated, requiring from developers only the initial guidelines and a slight touch of post-processing.

As a source data format, I use LaTeX code for linguistic examples; see Fig. 31, which displays the source code for (125). However, if one would like to make a similar project or add more data to an existing one, one could use any other format of examples provided by another tool (such as FLEx or Field Linguist's Toolbox), or a tab-separated text file.

(125) nēnu rāmu-cēta kāru-∅ kaḍig-inc-ǣ-nu — Telugu
I Ramu-INS car-ACC wash-CAUS-PST-1SG
'I had the car washed by Ramu'/ 'I caused the car to be washed by Ramu.'
Krishnamurti 2003, p. 433, (20f)

As shown in Fig. 31, the LaTeX examples I use are realized with the ExPex package[39].

[38] https://software.sil.org/software-products/, accessed on 01.05.2026
[39] https://ctan.org/pkg/expex, accessed on 01.05.2026

```
\pex \label{TEL_Caus_Ins}
\begingl
\gla ne\Uchar"0304nu ra\Uchar"0304mu-ce\Uchar"0304ta
      ka\Uchar"0304ru-$\varnothing$ kad\Uchar"0323ig-inc-\Uchar"01E3-nu//
\glb I Ramu-\Ins{} car-\Acc{} wash-\Caus{}-\Pst{}-\First{}\Sg{}//
\glft 'I caused the car to be washed by Ramu.'//
\endgl
\xe
```

Fig. 31. Example (21) as source LaTeXcode

Most of the Unicode characters are inserted via the `Uchar` command to ensure correct typing independently from the LaTeX editor. As for the glosses, most abbreviations are taken from the Leipzig package[40], and missing morphemes have been added by hand.

One of the assets of this data format is the already specified alignment between morphemes and glosses. Moreover, some information about morpheme types can be deduced from the labels themselves or from the descriptions provided by the Leipzig package.

I decided to use the ExPex + Leipzig example setup to automatically generate the lexical files for my metagrammar. The resulting script is given in Appendix D. It creates Morpho and Lemma files simultaneously, automatically detecting the example's language from the label (see `label` in Fig. 31). The script can take as many examples per language as needed. All non-ASCII symbols are typed as Unicode characters (as they appear) because they are supported by the XMG compilers, the TuLiPa parser, and the visualizer.

The transcriptions found in external sources (grammars, papers, etc.) are mainly left intact, unless overly complex diacritics compromise the legibility of the final diagrams (trees and frames). In those cases, simplifications are made provided they do not create additional ambiguities. Thus, the Kabardian (94) verb *q'ʷət* 'cut' is written in `Morpho_KBD.mg` as *q'wət*, which is perhaps less faithful to the real pronunciation, but has no decisive difference within the data used for the present metagrammar.

The automatically created `.mg` files require some post-processing from the metagrammar developers. Once the grammar is upscaled, further modifications can be introduced to the script to make its output cleaner and better compatible with future versions of the lexical files. At this stage of the project, it is an acceptable trade-off between the time required for grammar development and the time spent creating the script.

[40] `https://ctan.org/pkg/leipzig`, accessed on 01.05.2026

```
1 class CaseMarker_ti
2 import Abstr_CaseMarker[]
3 {<morpho>{
4   {morph <- "ti"|morph <- "ni"};
5   lemma <- "Acc";
6   cat <- casemark;
7   case <- acc  }}
```

Snippet 7: The `morpho` class showing the Tuvinian ACC marker

10.3 Lexical Resources

10.3.1 Morpho

The `morpho` dimension works with the actual language data. Snippet 7 demonstrates a sample `morpho` class, using Tuvinian accusative marker as an example. One can see that the body of the class (lines 4–7) presents a set of features relevant to the morpheme described by the class. All features are given as attribute-value pairs connected with a “<-” sign.

Two features are obligatory for the models to function: `morph` and `lemma`. They are not declared in the general preamble of the metagrammar because their definition is already part of the language. More details about these features are given in Section 10.3.1.1 and Section 10.3.1.2.

Other features needed for syntactic trees can be assigned to morphological entries, as in the current project. They are described in Section 10.3.1.3.1, Section 10.3.1.3.2, Section 10.3.1.3.3 respectively.

Snippet 7 lists one more feature: `cat`, standing for *category* of the node (roughly corresponding to the type of syntactic constituent or part of speech). This feature is discussed in detail in Section 10.3.2.2 in relation to the `lemma` dimension because it has crucial importance for linking `lemma` to `syn`. In the `morpho` dimension, the feature `cat` is used mainly for convenience.

10.3.1.1 Morph

The `morph` attribute is the essence of the `morpho` dimension since its value has to match one of the morphemes (non-whitespace symbol sequences) in the input line of the parser. Presumably, each non-ambiguous morpheme is listed once in the Morpho file for a language as the value of the `morph` attribute. Of course, one file can contain two or even more identical `morph` lines if two different morphemes look the same. Say, English *-s* in both *bananas* and *develops* will anyway be encoded as `morph <- "s"`, but in different classes. Thus, other attributes would help to distinguish between them (see below).

It also happens that the same morpheme can be realized in more than one way in a given language. For instance, Snippet 7 lists two allomorphs (variants) for the Tuvinian accusative marker. The disjunction operator indicates that either one attribute-value pair or the other can be used in a sentence. With this description, both allomorphs are aligned with the same lemma and the same feature structure. The compiler creates two models from this class, one for each morph variant.

10.3.1.2 Lemma

The `lemma` attribute contains the translation of the morph, either as an English (or any other language of linguistic description) word or a Leipzig-formatted gloss. The value of this attribute is also a string, no format restrictions apply. The purpose of the `lemma` attribute in the `morpho` dimension is to relate the class in question to the `lemma` dimension (see Section 10.3.2). This relation is the key to all lexical linking, since it ensures that the morpheme becomes visible to the syntax. As a result, `morph` is implemented in the syntactic structure together with all its features.

10.3.1.3 Special features

The three features above are required for all classes in the `morpho` dimension. However, some morphemes have additional features relevant to the overall description (as well as to parsing). Since the purpose of the current metagrammar is quite narrow, only the most necessary minimum of these additional features has been introduced. In other projects, the number of these features can be increased to meet their goals.

10.3.1.3.1 Case affixes: feature `case` As shown in Snippet 7, classes describing case morphemes have an additional feature `case`. This is a typed feature, which means that its value cannot be a free-form string, but is selected from a list of acceptable values. Those lists of values for each type are defined in the *preamble* of the metagrammar.

The feature `case` exists in all dimensions, and everywhere the list of acceptable values is the same. Thus, the real case found in a sentence becomes visible through the `lemma` dimension to `syn` dimension, where it receives its linguistic analysis.

10.3.1.3.2 Verbal roots: feature `trans` Verbal roots are marked for their lexical transitivity. This feature determines the shape of the future syntactic tree, telling how many branches of what kind it should take. This is what the feature `trans` stands for. In my implementation, I use a very simple set of values: `intr`, `tr`, and `ditr` (corresponding to *intransitive*, *transitive*, and *ditransitive*, respectively). These are thus the only three values allowed by the type of this feature. In the current version of the metagrammar, only transitive verbs are used in test sentences; thus, only transitive verbal roots are described in the lexical files. The other two values are provided to illustrate the project's potential scalability.

One could have set the feature `trans` to have integer (numerical) values from 1 to 3. However, since arithmetical operations are very limited (if not inexistent) in XMG, there is not much difference between numerical and categorical types.

10.3.1.3.3 Causative affix: features `caus` and `valtype` Additional features are also defined for the `morpho` class describing causative affixes. In the current project, there are two of those, but in fact, both would play a more important role once other constructions are added into the same metagrammar. First, I define the boolean feature `caus`, which is set to `yes` for the causative morpheme. As a result, all trees where the causative affix appears receive the positive `caus` value and thus constitute the subset of causative constructions within the whole metagrammar. One can interpret this boolean feature as a question: "Is this unit causative?", where a *unit* can be a morpheme, a verb, a sentence fragment, or a full construction.

The `caus` feature is similar to the `case` feature because it encodes the precise meaning of the morpheme. It may seem, however, that the `caus` structure being boolean is structurally different from the typed `case` feature. In fact, it is realized differently, but it functions in a similar way. The causative morpheme in all the languages of the sample is agglutinative. It means that if it is present in a word, the verb is causative, otherwise it is non-causative. So, the verb is never totally unmarked for causativity. The absence of the morpheme is just a way of assigning a negative value to the attribute. To continue the parallel with nominal cases, they also can be 'unmarked' in many languages, but it does not mean that nominals without overt case morphemes do not have any case. Instead, the absence of a segmental case morpheme assigns one particular value (the case that is allowed to be realized without an overt morpheme in the given language) to the respective attribute. Thus, despite different technical types, `case` and `caus` features exhibit some behavioral properties.

What makes the feature `caus` particular is its assignment not only to nodes of the syntactic structure, but also to free variables, both within and outside dimensions. For instance, the class `All Range Causative Constructions` starts with the statement `?Caus = yes` before even the syntactic dimension opens. The `?Caus` variable in this line is free and not typed, but it unifies with another variable with the same name used as a placeholder value for the `caus` attribute in a feature structure: `[caus = ?Caus]`. In this design, the value of the free variable `?Caus` unifies with the value of the feature `caus` coming from the morpheme.

Statements like `?Caus = yes` are useful for class categorization. Basically, this instruction means that all the dimensions of this class describe causative constructions exclusively. Further expansions of the metagrammar would comprise additional classes, following a similar guideline (e.g., `?Applicative = yes`). The idea is to delimit constructions that require different tree shapes from one another. Thus, grammar categories influencing the tree shape, mainly verbal derivations, would require features like `caus`, `appl`, and the like.

The second additional feature (attribute-value pair) assigned to causative morphemes in the lexicon is `valtype <- valincr`, which stands for *valency (changing) type = valency increase*. This, in contrast, does not correspond to any specific grammatical category but is rather conditioned by the architecture of the metagrammar.

Since XMG is not Turing-complete, its arithmetic capabilities are very limited. Namely, if one needs to encode the increase of valency by one, one cannot write something like *`valency += 1`. Instead, one needs to provide a cascade of conjunctions relating the initial value to the +1 value. (Luckily, the number of arguments a verb can take is very limited.)

To keep the main classes of the metagrammar clean and well-readable, I decided to create a special class in which the verb's transitivity is mapped to a new feature: the valency of the core. Thus, the feature `valtype` tells which type of valency-changing operation (if any) the morpheme represents. And its value, in terms of metagrammar architecture, determines the additional class to be used in the final construction. Currently, only one value of the attribute is supported, `valincr` (valency increase). Another possible value would be `valred` (valency reduction) once the metagrammar is expanded.

Constructional classes using the feature `valtype` are presented and discussed in Section 10.4.3.4.

10.3.1.4 Output

The output of the `mph` compiler after processing a `Morpho.mg` file is a set of feature structures stored in `Morpho.xml`. As discussed above, each allomorph receives its own feature structure. So, the number of output models corresponds to the number of the valuated classes plus the number of disjunctions. All the attributes of each feature structure become visible to other dimensions once the `morpho` dimension gets linked to the `lemma` and then to `synframe`. This is done by the parser, not by XMG.

10.3.2 Lemma

The lemma dimension is perhaps the key to effective parsing, since it is where Lexicon and Syntax connect. The purpose of the Lemma dimension is to attach each morpheme to the correct type of syntactic trees and make sure that this link is unique. Given this interface-like nature of Lemma – especially in a project working primarily with morphologically rich languages and thus having a well-developed Morpho dimension – all lemma classes in the presented solution have exactly the same structure, see Snippet 8.

The feature `entry` (see line 3 in Snippet 8) serves for unifying two dimensions:

```
1 class Lemma_CausAffix{
2 <lemma>{
3   entry <- "CAUS";
4   cat <- causaff;
5   fam <- AnchorCausAffix
6 } }
```

Snippet 8: The lemma class for a causative affix

lemma and morpho. It is very special since no feature of the same purpose exists between any other pair of dimensions. Its realization is also quite peculiar: usually, unification is possible only of features with the same names. However, the entry feature in the lemma dimension unifies with the lemma feature in the morpho dimension.

The feature cat (see line 4 in Snippet 8) gives some information about the contents of a morpheme; more precisely, something about its usage in a real language. The features of the same kind are, for example, case or trans (in *morpho* and *syn* dimensions).

The last feature fam (line 5 in Snippet 8) provides the parser with instructions on how to use a given model. In other words, it binds the lemma class to a specific set of syntactic trees (tree fragments).

The sections below provide more details on each feature in the lemma dimension.

10.3.2.1 Entry

The attribute entry takes a string that is supposed to be *the lemma* of a given word. Put differently, the entry is the morpheme's identifier in the lemma dimension.

As already stated, I use English translations as lemmas for lexical roots, and Leipzig glosses as lemmas for grammatical morphemes. This approach is not the only possible one. For example, in a metagrammar of one language, one could use the base form or a form traditionally used in dictionaries as the value of this attribute. Another option is to use computer-generated identifiers that ensure value uniqueness.

To ensure correct linking between lemma and morpho dimensions, the entry must take the same value as the lemma in the morpho dimension. Since these features are not typed, human attention or an additional machine check-up is crucial to prevent errors.

10.3.2.2 Category

The attribute cat stands for *category* and takes the name of a syntactic category to which this lemma belongs. As Snippet 8 shows, the lemma spelled as "CAUS" belongs to the category causaff (which stands for *causative affix*). This means that the lemma spelled as "CAUS" could be anchored only to syntactic nodes that have [cat = causaff].

In a very general understanding, in a project without the morpho dimension, the values of cat would be part-of-speech tags with the required degree of specificity (coarse-grained vs. fine-grained). In a metagrammar, whose unit is a morpheme, types of morphemes would be set as cat values.

The research project's focus also influences the definition of cat values. Namely, in my metagrammar, the causative affix is a very important morpheme and hence receives a special category causaff. The same logic led to the introduction of a separate category caseaff for case affixes. In contrast, all other verbal and nominal grammatical morphemes receive categories aff and naff, respectively.

The feature cat is used in other dimensions, too. In the morpho dimension, it is used similarly to indicate the category of the morpheme and the tree where it can anchor. However, the specification of the feature cat in the morpho dimension is not necessary.

In the syn dimension, categories are usually assigned to all nodes. These indicate not only the places where lemmas can attach but also other layers of the clause.

It is important to highlight that the value of cat is not a free string but a typed value. In the preamble of the main metagrammar, there is a type CAT defined, which takes a closed set of values. In order to be compatible with the rest of the metagrammar, lemma classes must have one of those as the value of cat. The list of possible values is given in Snippet 9.

```
type CAT = {sentence, clause, core, ecs, np, core_n,
    nuc_n, noun, arg, nuc, verb, nroot, vroot, causaff,
    casemark, aff, stem, argaff, naff}
```

Snippet 9: Values of type CAT given in the preamble

10.3.2.3 Family

The third attribute listed in Snippet 8 is fam, which stands for *family*. The XMG documentation[41] defines it as "TAG family which can use this lemma as anchor". As one can see from the citation, the notion of the family comes from the TAG formalism but is developed in the TAG-RRG-XMG framework I use. In the present work, a *family* is a class that is valuated by the synframe dimension and thus creates a set of trees available for the parser.

Being assigned to a family means for the lemma being assigned to a set of trees in the grammar. In this respect, family is an auxiliary attribute that serves to connect dimensions but plays no particular role within a single dimension. It functions exactly like lemma, but on a higher level. The lemma attribute ensures the transfer of all the

[41] https://github.com/spetitjean/XMG-2/wiki, accessed on 01.05.2026

information contained in the morpho dimension to the lemma dimension, while the family ensures the transfer of all that information to the syn (syntactic) dimension.

The family feature also ensures that each lemma class appears in a single type of tree. Thus, even if two lemmas have the same category but have different syntactic behavior, they would belong to different families. This functionality can be used to reflect some linguistic analyses. For example, some XMG projects use families to distinguish between intransitive and transitive verbs so that they do not take the inappropriate number of arguments.

In the current metagrammar, one purpose of the family feature is to control the language selection. In other words, it ensures that sentences in a given language are not parsed with the metagrammar describing another language. Thus, nominal roots belong to families Argument_LAN, where LAN stands for the language name. It means that it is technically impossible to parse a noun phrase in one language with a tree belonging to another language, even accidentally. Similarly, all verbal roots belong to the language-specific families Intersection_LAN. (More details about the name of this family and the way it is realized can be found in Section 10.4.6.) Although the Intersection_LAN class describes the whole construction made of a causative verb and three NPs, it is set as a family for one single morpheme – the verbal root. The reason is that my metagrammar follows LTAG guidelines and has exactly one lexical anchor per tree. Therefore, in the elementary tree describing the whole sentence, only one lemma – namely, the verbal root – is anchored directly. All the other morphemes anchor to their respective elementary trees belonging to other families and are combined with trees generated by the Intersection class via parsing operations like substitution or sister adjunction.

The approach used for root morphemes is, nevertheless, not the most optimal one for relating grammatical morphemes to families. From a tree-building perspective, all grammatical morphemes need just one syntactic node that would anchor a lemma and make all its features and the features of the respective morph accessible from the tree templates generated by the <syn> dimension. The disambiguation and the control of the correct anchoring are thus due to the cat attribute that selects the correct substitution node to which the small tree would attach. (See more about the anchoring of grammatical morphemes in Section 10.4.1.) With this architectural similarity in mind, I decided to create families that anchor respective grammatical morphemes, without language differentiation. If two morphemes of a single language look alike, they are distinguished by their lemma and cat values. If two grammatical morphemes of different languages look alike, there is no need to disambiguate them since they would anyway attach to root morphemes (whose disambiguation is controlled through LAN families). Since I am not dealing with sentences written in more than one language double-checking for ambiguity seems unnecessary.

10.3.2.4 Final remarks

To sum up, each Lemma class has an identifier and two attributes that ensure the correct placement of the lemma: `fam` selects the tree, and `cat` indicates the correct position within the tree. To avoid repetitions and errors, one could encode the common parts of all lemma classes separately in order to import them into target classes afterward. Necessary specifications can be added in the final classes. I call these common parts *abstractions*; see Snippet 10, which features all of them. These very same abstractions are used for all morphemes in all languages. The only difference is the language-code placeholder (see `LAN` in Lines 3 and 7), which is substituted for a language identifier (usually an ISO 639-3 code) in each file.

Note that all abstractions, excluding the last one, are pairs of `cat` and `fam`. The third required field `entry` is obviously different for each class. Thus, it needs to be specified in each class corresponding to a lemma, while `cat` and `fam` can be imported from an abstraction. This process is shown in Snippet 11. The import mechanism in XMG is implemented as a simple conjunction: it reads all the instructions from all sources (imported classes) as if they were written in a single class, one after another. Therefore, one can just write the entry line in the class without any further complications. All other nominal root classes in the same language would be organized in the same way by importing the same abstraction for the `NRoot` and specifying the entry.

In order to become available for the parser, Lemma classes have to be compiled with a specifically designed `lex` compiler. To become compiled, each class has to be valuated, as well as in all the other XMG dimensions. Since no disjunction is used in the Lemma classes, each class outputs exactly one model. Thus, the resulting .xml file that is fed to the parser comprises as many models as there are morphemes in the sentences one would like to parse in a given language.

10.4 Construction classes

The term "Construction classes" refers to all classes that are responsible for describing constructions. These are all classes described in the main metagrammar, excluding the Language Plugins. In other words, these are the classes compiled with the `synframe` compiler. This section presents all of them in a bottom-up order: from classes describing single morphemes in the syntactic and semantic perspective to the classes describing whole sentences. Most classes are reused across languages; however, upper-level classes differ between head-marking and dependent-marking languages. These are described in due places throughout this chapter.

```
class Abstr_Lemma_NRoot{
<lemma>{cat <- nroot;
        fam <- ArgumentLAN  }}

class Abstr_Lemma_VRoot{
<lemma>{cat <- vroot;
        fam <- IntersectionLAN  }}

class Abstr_Lemma_CaseMarker{
<lemma>{cat <- casemark;
        fam <- AnchorCaseAffix  }}

class Abstr_Lemma_VInflect{
<lemma>{cat <- aff;
        fam <- AnchorAdjunctVerbalInflectAffix  }}

class Abstr_Lemma_NInflect{
<lemma>{cat <- naff;
        fam <- AnchorAdjunctNominalNoncaseAffix  }}

class Abstr_Lemma_ArgAffix{
<lemma>{cat <- argaff;
        fam <- AnchorArgumentAffix  }}

class Lemma_CausAffix{
<lemma>{
  entry <- "CAUS";
  cat <- causaff;
  fam <- AnchorCausAffix  }}
```

Snippet 10: Abstractions for Lemma classes

```
class Lemma_NRoot_book
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "book" }}
```

Snippet 11: Importing an abstraction and specifying the entry of a Lemma class

10.4.1 Anchoring

An anchor is an important notion in a metagrammar and also in the subsequent parsing. The concept of a lexicalized grammar suggests that every elementary tree (i. e., every tree that is finally compiled) has exactly one anchor. Other tree fragments that are used exclusively as imported blocks within elementary trees can be non-lexicalized, i.e., have no anchor.

When dealing with real language (even, as in the present work, with quite short and mostly elicited sentences), the solution respecting this requirement is sometimes not quite straightforward. Therefore, in this section, I present the anchoring schemes used for different morpheme types.

In the target sentences of my sample, there are two main classes of words: verbs and nouns. Each of these classes has morphemes of two kinds: lexical roots and other

```
class AnchorOneMorpheme
{<syn>{ node (mark = anchor)[] }}
```

Snippet 12: The class for anchoring a single morpheme

grammatical morphemes. Since roots contribute the lexical meaning to the whole word, they are used as anchors for the respective classes. Other morphemes are added using parsing operations. The remainder of this section presents the details of this process.

Verbal roots are very important in any sentence, and especially in the present work. In particular, they have lexical features that appear to be important for the structure of the whole sentence. Moreover, the frame of the verb determines the appearance of the sentence tree; hence, there must be a special linking between the `syn` and `frame` dimensions. For all these reasons, a special class is created that marks a node of category `vroot` as the anchor. Thus, the lemma attaches directly to it. In the same class, the verbal frame is instantiated, see details in Section 10.4.3.1. Note that in order to keep all the higher levels of the sentence structure (e.g., `VerbalSpine`, see Section 10.4.3.5) lexicalized, the class `VerbalRoot` is imported into them. Thus, all these classes have the same anchor, which remains unique per class.

The same approach is used for anchoring nominal roots. Nominal roots are anchored to a class that combines all of the information concerning the processing of nominals, `class NounWithCase`, see Section 10.4.2.1. The lexicalized class `class NounWithCase` is later imported into the class `ArgumentLAN` (where `LAN` stands for language name) and yields a new lexicalized class having the node `?NROOT` as its anchor.

Throughout my test data, there is one type of verbal morpheme that is obligatory for all words: a causative affix. Similarly, one class of grammatical morphemes is obligatory for nominals: the case marker (the lack of overt case marking is handled, see details in Section 10.4.2.1). Thus, in the classes describing words (verbs and nouns), there is a substitution node with the category of the morpheme that is awaited there (`causaff` in verbs, `caseaff` in nouns). When it comes to anchoring these morphemes, the only thing required is to create an elementary tree consisting of a single anchor node. Thus, the briefest possible class is created, see Snippet 12. Importantly, it does not specify the node category. Thanks to that, both case affixes and causative affixes can anchor to it.

Theoretically speaking, morphemes of other categories that already exist in the metagrammar (say, roots) could have been anchored to this class as well. In fact, they do not because it is prohibited for their lemmas. The `fam` parameter of the lemma specifies the highest class to which a morpheme can anchor, see Section 10.3.2.3. Since the class `Anchor One Morpheme` is listed as the family only for case and causative affixes, only they anchor to it in the presented solution. In case the metagrammar is expanded to handle other types of morphemes, no new classes for anchoring them would be re-

```
class NounWithCase
  export ?NodeNoun ?Case ?LanguageFeatures
  declare ?NodeNoun ?NodeNRoot ?NodeCaseMarker ?Case
      ?LanguageFeatures
  { <syn> {
    node ?NodeNoun [cat = noun, case = ?Case];
    node ?NodeNRoot (mark = anchor) [cat = nroot];
    ?NodeNoun -> ?NodeNRoot;
    {
      { node ?NodeCaseMarker (mark = subst) [cat =
          casemark, case = ?Case];
      ?NodeNoun -> ?NodeCaseMarker; ?NodeNRoot >>
          ?NodeCaseMarker}
    |
      { ?Case = ?LanguageFeatures.UnmarkedCases }
    }
} }
```

Snippet 13: The class describing nominals

quired. They could still be anchored to the class `Anchor One Morpheme`, one just needs to specify this in the `lemma` requirements.

There are other morphemes that are described in the present metagrammar. In addition to derivational morphemes (such as the causative affix), verbs usually have quite a range of inflectional morphemes: tense, person, etc. These morphemes do not influence tree shape and are not important for the analysis and representation of causative constructions. Nevertheless, they are present in real language data, and their absence makes sentences ungrammatical. As a shortcut for processing verbal inflectional affixes, I decided to treat them as adjuncts at the level of the verb, see details in Section 10.4.3.3. This approach has an important benefit: adjuncts can attach to the main tree in any order; therefore, I do not need to describe the morphological rules of inflection for each language (which is not the focus of the present work).

A tree for anchoring adjuncts should contain at least two nodes: one for the head of the adjunct and the other indicating the category to which it should adjoin. Thus, there are two identical trees for verbal and nominal adjuncts, respectively (see Fig. 32).

a. Verbal b. Nominal (non-case)

Fig. 32. Anchoring inflectional morphemes

10.4.2 Nominals

10.4.2.1 Noun with case

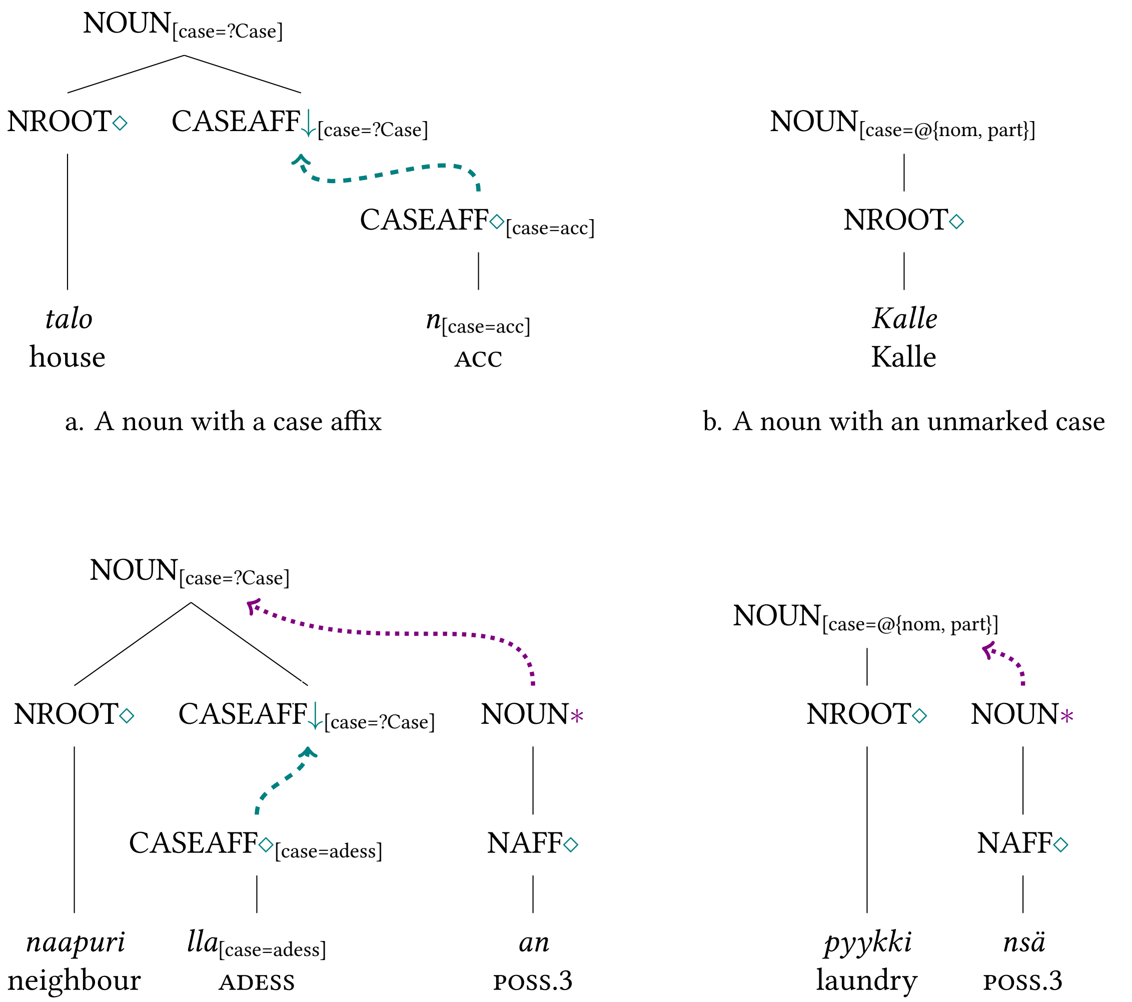
To parse a noun, one needs to assemble several morphemes that constitute a single graphical noun. In my metagrammar, there are three types of morphemes (categories) related to nominals: `nroot`, which stands for *nominal root*, `caseaff`, which stands for *case affix*, and `naff`, which stands for *nominal (inflectional non-case) affix*.

The main class for describing nouns in dependent-marking languages is the class `Noun With Case`, see Snippet 13. The class `Noun With Case` starts with declaring the node of category `noun` (line 5). It means the whole class is responsible for building a single noun. Apart from the category, it has one feature assigned, `case`. As its value, it takes the variable `?Case`. Once the value of the feature `case` is assigned by the construction or comes from the lexicon, the variable gets replaced by it. Further syntactic positioning of this word is described in the class `Nominal Spine` (see Section 10.4.2.3).

The node `?NodeNoun` of category `noun` governs another node, `?NodeNRoot` of category `nroot`, which stands for *nominal root*, see lines 6-7. Thus, every noun described by this metagrammar has to contain a root. The node `?NodeNRoot` is also an anchor, therefore, lemmas from the lexicon attach directly here. Note that, according to the language reality, the node for the nominal root is not marked for case.

While the elements described in lines 5-7 are necessary for any noun, lines 8-13 present a disjunction of two options that show the presence and the absence of an explicit case affix on the noun. The first option is shown in lines 9-10. The node `?NodeCaseMark` creates the place for the case affix to attach. Since there is already one anchor in the tree, the case morpheme cannot anchor to it directly, given the requirement for lexicalization. Therefore, the node is marked as a *substitution node* and opens the position for a smaller tree created by the class `Anchor One Family`. The value for the feature `case` thus unifies with the one from the `morph` and also with the one assigned to the node `?NodeNoun`. Line 10 positions the case affix in the tree. Namely, it tells that the case node is governed by the `?NodeNoun` and preceded by the `?NodeNRoot`. This last instruction is not necessarily required. Once there is a language in the sample that has case prefixes instead of case suffixes, it will not be valid. However, this precedence requirement is kept in the current version of the metagrammar for faster compilation.

The second option (line 12) describes nouns without any overt case marker. In the absence of an overt case morpheme, the value of the case feature needs to be assigned to the noun as a whole word. Therefore, line 12 contains instructions about the variable `?Case` without declaring additional nodes. Since there is no case morpheme to bring the value of this feature from the lexicon, it is taken directly from the language plugin (represented as the variable `?LanguageFeatures`, see Section 10.5.1). The feature `UnmarkedCases` and its structure are described in detail in Section 10.5.2.



a. A noun with a case affix

b. A noun with an unmarked case

c. A noun with case and possessive affixes

d. A noun with an unmarked case and a possessive affix

Fig. 33. Models generated by the class `NounWithCase` and the corresponding Finnish words

Usually, languages have several cases that can remain unmarked. In this situation, the value of the variable `?Case` within this class remains undetermined (i.e., the atomic disjunction remains unresolved) as long as further requirements are not introduced at the level of constructions, which are compiled by a different class. As a result, the class generates two models, which are illustrated in Fig. 33a and Fig. 33b. The underspecified case value will be, however, specified during parsing, when the noun with a fixed case gets substituted in an argument position under the CORE.

10.4.2.2 Nominal inflection

As for other inflectional morphemes, they are anchored to separate trees rooted in the node of category `noun` (see Fig. 32b). The `noun` node in this tree is marked with a "*", which means that the tree depicted in Fig. 32b does not function independently but must attach to another tree containing a `noun` node through the operation of *sister adjunction*. This operation takes place while parsing. For that reason, one cannot use any features of trees like Fig. 32b within the metagrammar. The current project has been designed in such a way that the category `naff` (non-case inflectional nominal affix) is given only to those morphemes that have no relevant features for the metagrammar of causatives. Therefore, they can easily be added at a later stage. This shortcut can be used in other projects for some morphemes that are not of primary interest. On the other hand, if one is interested in any other nominal features except for the case, one could encode the so-called *other nominal inflectional morphemes* within the metagrammar (like case morphemes are handled in the current project).

The derivation of trees using the tree fragments generated by the class `Noun With Case` and other classes where nominal morphemes anchor are shown in Fig. 33c and Fig. 33d. The operation of sister adjunction is symbolized by the dotted arrow. Although in all Finnish words in Fig. 33, non-case morphemes follow case markers, there is no technical requirement for a specific linear order. Sister adjunction is not sensitive to the rules of precedence defined in the metagrammar (because it is a parsing operation); therefore, adjunct morphemes can appear anywhere in the tree generated by the `Noun With Case` class. If one needs to add constraints to adjunct morphemes (e.g., if one has both prefixes and suffixes that need to be distinguished from each other), one would need to define edge features (not used in the current metagrammar).

10.4.2.3 Nominal Spine

Arguments are nominal phrases. In the current metagrammar, nominal phrases are modeled according to the layered structure of the clause in RRG but follow the newer notation used by the TreeGraSP project[42]. The class `Nominal Spine` creates all the syntactic nodes: `?NP`, `?CORE_N`, `?NUC_N`. These nodes are in dominance relation, see Fig. 34. At each of these nodes, modifiers (determiners, adjectives, and others) can adjoin.

All these nodes are decorated with features. The first of the features is `cat`, the syntactic category. As already demonstrated in Section 10.3.2.2, this feature determines the type of node. In the syntactic dimension, it also determines the type of adjunct trees that could attach to each given node.

The second feature is `case`, as introduced in Section 10.3.1.3.1. It percolates from the case morpheme up to the highest level, i.e., `?NP`. Thus, it is accessible at any layer

[42] `https://treegrasp.phil.hhu.de/`, accessed on 01.05.2026

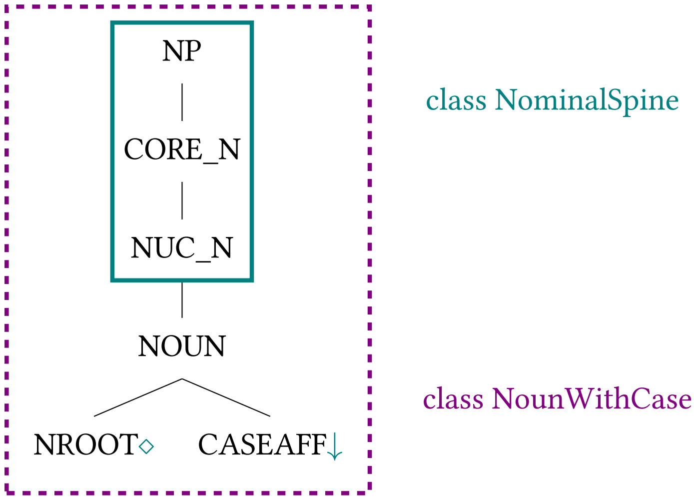


Fig. 34. The elementary tree for arguments produced by the class `Noun With Case` (with the class `Nominal Spine` imported)

of the Nominal Spine and can therefore participate in agreement and other connections with all possible adjuncts. At the `NP` level, the `case` feature is visible to the higher levels and thus can be controlled by other classes.

The feature `mr` stands for *macrorole* and is assigned to `NP` nodes by a higher-level class (see Section 10.4.5). Therefore, it does not appear on the lower levels.

The last feature assigned to the layers of the Nominal Spine is `i`, which stands for *identifier*. This label is assigned to the NP and ensures its correct linking to the semantic structure (frame).

The important fact about all these features is that they share the same value. Thanks to this, the feature values unify at all levels and across all dimensions. On the other hand, the feature values assigned to one NP would differ from those assigned to another NP, preventing different NPs from mixing. See the model produced by the class `Nominal Spine` (when it is compiled as a standalone class) with all the features in Fig. 35.

However, in the solution I present, the class `Nominal Spine` is not compiled alone but imported into another class, `Noun With Case` (described above in Section 10.4.2.1). Thus, everything contained in the `Nominal Spine` class appears in the `Noun With Case` class (see details on import in Section 9.1.1). This is represented in Fig. 34 with rectangle boxes: all the nodes that are declared in the class `Nominal Spine` (solid line box) are also contained in the class `Noun With Case` (dashed line box). As a result, although the class `Nominal Spine` is not lexicalized itself, it is never compiled alone but always as an extension of the lexicalized class `Noun With Case`.

The main asset of this architecture is its versatility in combining the spine with various types of elements. In addition to nouns, other elements can also serve as the morphological heads of nominal phrases (or, in classical RRG terms, reference phrases). There can also be pronouns, clausal elements, etc. (not to mention more exotic phenom-

$$\begin{bmatrix} \text{case} & \boxed{V2} \\ \text{cat} & \text{np} \\ \text{e} & \boxed{V3} \\ \text{mr} & \boxed{V4} \end{bmatrix}$$

$$\Big|$$

$$\begin{bmatrix} \text{case} & \boxed{V2} \\ \text{cat} & \text{core\_n} \\ \text{e} & \boxed{V3} \end{bmatrix}$$

$$\Big|$$

$$\begin{bmatrix} \text{case} & \boxed{V2} \\ \text{cat} & \text{nuc\_n} \\ \text{e} & \boxed{V3} \end{bmatrix}$$

Fig. 35. The non-lexicalized model produced by the class `Nominal Spine` (without being imported elsewhere)

ena). For sure, each of these elements would require a separate class that would anchor all the necessary morphemes and combine them into a single word (similar to `Noun With Case`). For example, a pronominal phrase would still contain `?NP`, `?CORE_N`, `?NUC_N` (where *N* stands for *nominal* as a type of phrase, not for *noun*) and the new node `PRONOUN` instead of `NOUN`. Once one would like to create a class for pronominal arguments, one would need to define the node `PRONOUN` and everything below it, i. e., its morphemic structure. However, there is no need to define all the syntactic levels of the nominal phrase once again. They can be simply imported into any other class that is eligible for becoming a syntactic unit of NP. This requires minimal effort, saves development time, and prevents accidental errors.

Another alternative would have been to perform the `import` the other way round: instead of importing the `Nominal Spine` into the `Noun With Case`, import the `Noun With Case` into the `Nominal Spine`. This option is shown in Fig. 36. In this solution, the lower part, which is lexicalized, is imported into the upper part, overlapping with it. As a result, there is no non-lexicalized class. However, it is more difficult to handle different types of NPs with this solution, which makes it less desirable than the approach described above and depicted in Fig. 34.

There is another alternative architecture that does not use `import`. One could create the substitution node `NOUN`↓ within the class `Nominal Spine` and add the nodes below while parsing through the substitution operation. However, in this architecture, the class `Nominal Spine` would remain non-lexicalized. However, instead of remaining a tree fragment, it would require compilation. In contrast to the previous solution,

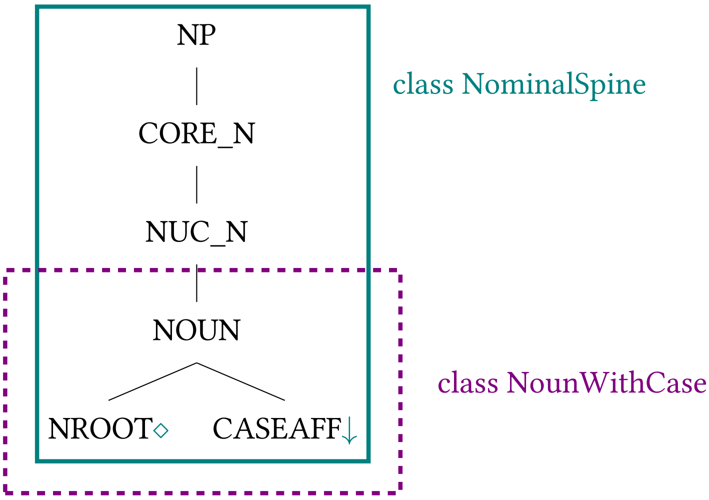


Fig. 36. An alternative import of the class `Noun With Case` into the class `Nominal Spine`

this approach is not welcomed in the formalism I use, namely, the lexicalized TWG.

Finally, the last alternative would be to refuse to separate information about NPs across multiple classes and just encode all nodes in the same class `NodeWithCase` one after another. Although it is the easiest way in the short term, it has a very low potential for further expansion (in comparison to the previous alternatives). Therefore, I prefer higher modularity for the sake of the combinatorial potential of the whole metagrammar.

Also, the separation at the NOUN level is linguistically salient. What is below NOUN belongs to the morphology, whereas the layers above it are syntactic. Thus, even in the absence of a separate morphosyntactic dimension, the metagrammar reflects well the levels of the language to which the described entities belong.

10.4.2.4 Argument in head-marking languages

For dependent-marking languages, the class `Noun With Case` contains all the necessary information and thus is ready to be intersected with a language plugin (see details in Section 10.4.6). In head-marking languages, however, the situation is different and thus requires an additional class.

In head-marking languages, not the NPs, but the bound verbal affixes are proven to be core arguments, according to Section 2.4.5. The NPs, in turn, are placed in Extra-Core Slots (ECSs). Nevertheless, NPs need to keep reference to the verbal morphemes they are bound to. In other words, a tree representation must show which argument affix is lexically specified by which NP.

To ensure this reference, it would be nice to describe both the NP and the argument affix within the same tree. It is, however, not that easy since the nodes of an argument affix and an ECS are located in really different places of the tree, see Fig. 37. In similar situations, the wrapping substitution parsing operation has proven useful.

```
1  class ArgumentHM
2  export ?CLAUSE ?ARGAFF ?ECS ?NP ?ARG ?Case ?eargaff ?Mr
3  declare ?CLAUSE ?ARGAFF ?ECS ?NP ?ARG ?Case ?eargaff ?Mr
4  { <syn> {
5  node ?CLAUSE [cat = clause];
6  node ?ARG (mark = ddaughter)
7      [cat = argaff, case = ?Case, id = ?eargaff, mr =
          ?Mr];
8  node ?ARGAFF (mark = anchor)
9      [cat = argaff, case = ?Case, id = ?eargaff, mr =
          ?Mr];
10 ?ARG -> ?ARGAFF;
11 {
12     {node ?ECS [cat = ecs, id = ?eargaff];
13      node ?NP (mark = subst)
14          [cat = np, case = ?Case, id = ?eargaff];
              ?CLAUSE -> ?ARG;
15      ?CLAUSE -> ?ECS; ?ECS -> ?NP
16     }
17     |
18     {?CLAUSE -> ?ARG}
19 }  } }
```

Snippet 14: The class describing the wrapping substitution of arguments in head-marking languages

Thus, Laura Kallmeyer suggested using wrapping substitution to describe arguments in head-marking languages (Kallmeyer 2021). Rainer Osswald (p. c.) supported this idea. The key idea is to describe the argument affix and the ECS in a single tree so that they can share features. However, one of the edges of this tree is a *d-edge*, allowing the tree to split and wrap around another tree.

Below, I describe in detail how I implement wrapping substitution to describe arguments in head-marking languages. For doing so, I define a class `ArgumentHM` (where *HM* stands for *head-marking*), which is displayed in Snippet 14.

First, one needs to determine the node that serves as the head for both the ECS and the argument affixes. Since ECSs are extra-core slots and argument affixes are in-core entities, this node is `CLAUSE` as it spans over both core and non-core elements. The ECS is a usual node below CLAUSE level, thus it is directly dominated by the node `CLAUSE` and dominates the node `NP` in turn (see line 15 in Snippet 14). The `?NP` node is marked as a substitution node. It means that any `?NP` described by the class `NounWithCase` can become an Extra-Core Slot. The class `NounWithCase` describes nouns without overt case and nouns with an overt case morpheme, hence, it is suitable for constructions with any locus of marking. Namely, my solution can adequately describe sentences like Fig. 37, where there are both verbal affixes and case markers.

The argument affix, in contrast to the ECS, appears not directly below the clause but somewhere lower, as also shown in Fig. 37. For this reason, it cannot be directly

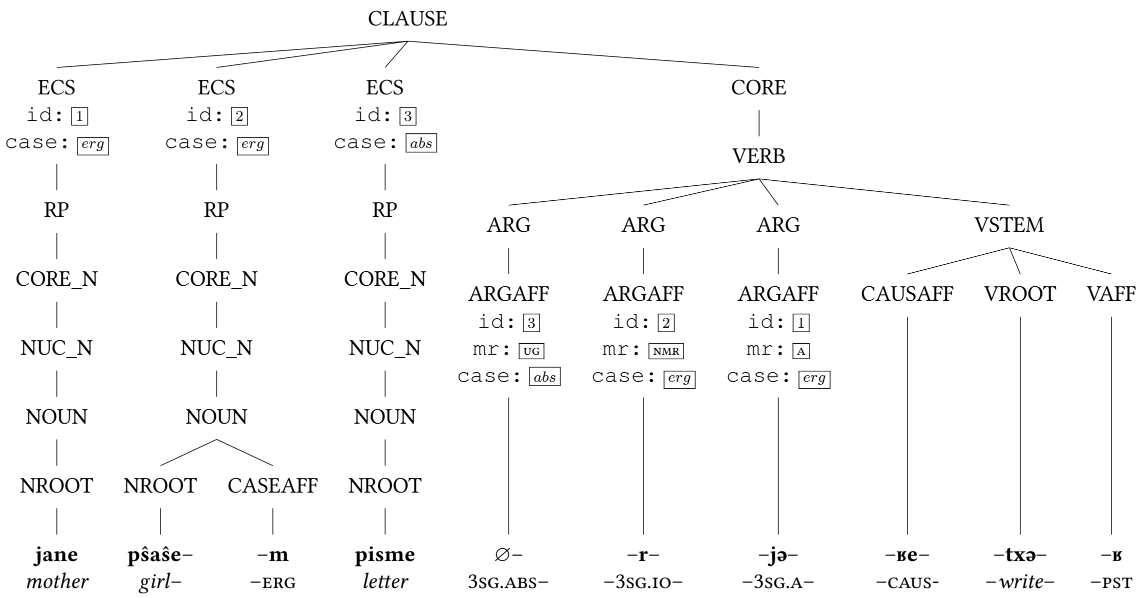


Fig. 37. The derived tree for the Adyghe sentence 'Mother made the girl write a letter.'

dominated by the CLAUSE in Snippet 14. Instead, the node `?ARG` (which stands for *argument*) is accompanied by the `(mark = ddaughter)`. It means that the edge leading from the `?CLAUSE` node to the `?ARG` node is a *d-edge*, i.e., the edge that gets replaced by all the nodes that appear in the tree between the `CLAUSE` and the actual place of the argument affix (see details in Section 8.2.2).

In Snippet 14, the node `?ARG` dominates the node `?ARGAFF` of the same category `[cat = argaff]` (cf. line 6 and line 8). The lower node, `?ARGAFF`, is marked as an anchor, meaning that this is the place where verbal morphemes denoting argument affixes anchor. This anchor makes the tree described by the `class ArgumentHM` lexicalized.

Conceptually, there is no need for two nodes of the category `[cat = argaff]`. Their doubling is due to a technical limitation: only one mark is allowed per node. Thus, if a node is both an anchor of a morpheme and a d-daughter of another node, it should have received two marks – `(mark = ddaughter)` and `(mark = anchor)` – which is technically impossible. Therefore, two nodes are created, each of them receiving a single mark. In the trees, this doubling has no negative effect because both nodes are in the same category and allow all features to percolate. This limitation of properties has been described in Section 9.3.

Since both the nodes `?ARGAFF` and `?NP` are in the same tree, they can share feature values. Namely, they share the feature `id`, which relates the noun to the respective verbal affix. The second important feature is `case`. It prevents the creation of incorrect links between NPs and verbal affixes; e.g., in Fig. 37, the noun 'letter' is never linked to the ∅ '3SG.ABS' affix. The feature `mr` (macrorole) is not shared by both nodes because only true arguments are assigned macroroles. Since ECSs are not arguments, they are

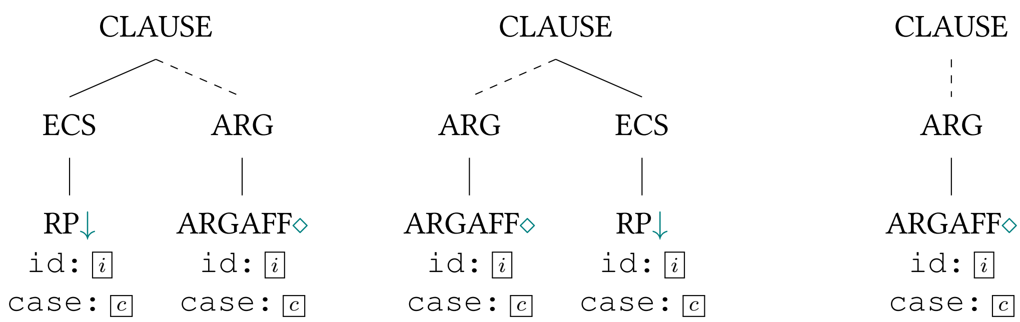


Fig. 38. Tree representations produced by the `class ArgumentHM` (feature structure simplified)

not eligible for sharing this feature.

Sentences in head-marking languages are grammatical without NPs. This option is also encoded in the `class ArgumentHM`. Lines 14-15 in Snippet 14 show a second disjoint option, where no `ECS` node appears in the tree. As a result, `class ArgumentHM` produces three trees displayed in Fig. 38: two trees with an ECS (whose position relative to the `?ARG` node can vary) and one tree without an ECS.

10.4.3 Verbs

Verbs are essential for building sentences and, hence, constructions. In the present metagrammar, I suggest a solution, which (1) compiles semantic frames of verbs based on the information stored in the Lexicon, (2) provides a derivation of the semantic frame due to the increase of valency, and (3) relates arguments to the empty slots in the frame. From the syntactic point of view, my solution offers a description of building blocks common to all language types that can be reused in other constructions if the metagrammar is expanded, and two target classes for describing causative constructions (one for dependent-marking and one for head-marking languages).

10.4.3.1 Verbal Root

The anchor in all constructions containing verbs is the verbal root. The metagrammar class for this morpheme is thus the most important for setting the feature structure for the whole future sentence where the verb would appear. In the current metagrammar, it is also the class with the best-developed frame dimension.

The code for the class `VerbalRoot` is shown in Snippet 15. In the syntactic dimension, there is one single node `?VROOT` of category `vroot` (which obviously stands for *verbal root*). This node is marked as the anchor, which means that verbal root lemmas attach directly to this tree. (In this respect, this class is similar to other morpheme-anchoring classes, see Section 10.4.1.)

Apart from the category, this node has other features assigned. The feature `trans`

```
1  class VerbalRoot
2    export  ?VROOT ?Trans ?F1 ?F2 ?F3 ?eroot
3    declare ?VROOT ?Trans ?F1 ?F2 ?F3 ?eroot
4    {<syn> {
5          node ?VROOT (mark = anchor)
6           [cat = vroot, trans = ?Trans, e = ?eroot]};
7    <frame>{
8          {?Trans = intr;
9          ?eroot[event,
10         participant: ?F1]
11         }  |
12         {?Trans = tr;
13         ?eroot[event,
14            agent: ?F1,
15            patient: ?F2]
16         }  |
17         {?Trans = ditr;
18         ?eroot[event,
19            agent: ?F1,
20            recipient: ?F2,
21            theme: ?F3]
22     }
23 } }
```

Snippet 15: The class describing verbal roots

(which stands for *lexical transitivity*) has already been described in Section 10.3.1.3.2. This is the crucial part of this class, since it determines the semantic frame used for the entire sentence. The value of the attribute `trans` in the syntactic dimension is set to a variable `?Trans` (see line 6 in Snippet 15). It means that the class `VerbalRoot` can anchor verbs of any transitivity. However, depending on the value of this attribute, different frames are linked to this node. The feature `[e = ?eroot]` stands for *event frame* and assigns a variable from the frame dimension (see lines 9, 13, 18 in Snippet 15) to the syntactic node (end of line 5).

The `frame` dimension in the class `VerbalRoot` contains three disjoint blocks that imitate the implication (see details in Section 9.4.5). Each disjoint block starts with the variable `?Trans` being assigned one of the values that are available for the feature `trans`. Thus, the same class can be used for anchoring verbs of any lexical transitivity.

The rest of each block describes the semantic frame, with the number of participants corresponding to the lexical transitivity of the verb. Given the focus of this project, frames are formulated in a very generic way in the current version of the metagrammar. More details can be added once the research focus expands or changes. Each block opens with a declaration of a variable `?eroot`: the identifier of the frame, which is also assigned as a value to the attribute `e` in the syntactic dimension. In Fig. 39, displaying the output of the `<syn>` and `<frame>` dimensions for transitive verbs, one can observe the label $\boxed{\mathit{Frame1}}$ in both parts.

After the identifier variable is declared, the frame is given in square brackets, starting

$$\text{VROOT}\diamond\begin{bmatrix}\text{cat} & \text{vroot}\\ \text{e} & \boxed{Frame1}\\ \text{trans} & \text{tr}\end{bmatrix}$$

a. Output of <syn> dimension

$$\boxed{Frame1}\begin{bmatrix}\text{event} & \\ \text{agent} & \boxed{V1}\\ \text{patient} & \boxed{V2}\end{bmatrix}$$

b. Output of <frame> dimension

Fig. 39. Outputs of the class `VerbalRoot` for transitive morphemes

with the type of event it describes. Since the current metagrammar gives just the most prototypical and general frames for verbal roots, all of them are called *event* (see lines 9, 13, 18 in Snippet 15). Afterward, all the participants are listed as attribute-value pairs. The attributes are general roles corresponding to each type of event. As values, they receive free variables which turn to labels in the output models (consider variables `?F1` and `?F2` in lines 14-15 in Snippet 15 and the labels $\boxed{V1}$ and $\boxed{V2}$ in Fig. 39b)[43].

In total, the class `VerbalRoot` generates three models, each comprising a syntactic and a frame component, which are linked through the shared value $\boxed{Frame1}$. The linking made within this class is very important for the whole syntactic structure, although it happens early on the morphological level. If one would like more elaborate frames, one could create several classes of this kind (using disjunction or otherwise), ultimately for each type of frame used in the metagrammar. No matter how many of those classes there are in the metagrammar, they all need to export the `?VROOT` node that is later imported to all other classes describing more complex morphological and syntactic structures.

10.4.3.2 Verbal Stem

In linguistics, a verbal stem is a morphological unit that includes the root and all derivational affixes but not the inflectional affixes. In a project dedicated to verbal derivation, this unit is of major importance. Therefore, in my metagrammar, there is a special class called `Causative Verbal Stem`, as shown in Snippet 16. Its function is threefold. First, it describes a syntactic structure in which the verbal root and the derivational affix combine to form a single unit. It is done in the `<syn>` dimension (see lines 6-7 in Snippet 16). The newly created node `?VSTEM` dominates both the `?VROOT` and the `?CAUSAFF` nodes and acquires all their features. To get access to the `?VROOT` and the `?CAUSAFF` nodes, the class `Causative Verbal Stem` imports the classes that contain the necessary nodes, namely, `VerbalRoot` and `CausAffix` (see line 2). Thanks to that, line 7 can simply name the nodes without specifying any further information about them. Note that no precedence constraints are given in line 7. So, the same class

[43] The variable naming might be somewhat confusing, though it is due to some historical reasons. In the code (e.g., Snippet 15), the `?F` names stand for something like *frame variables*, while in the output (e.g., Fig. 39), the $\boxed{V}$ names are used in all dimensions and stand for just *variables*.

```
1  class CausativeVerbalStem
2      import VerbalRoot[] CausAffix[] ValIncrease[]
           VerbalSpine[]
3      export ?VSTEM ?Trans ?Caus ?everb ?F0 ?F1 ?F2 ?F3
           ?Valnc ?Valtype ?V
4      declare ?VSTEM ?Trans ?Caus ?everb ?Valnc ?Valtype
           ?V
5      { <syn> {
6      node ?VSTEM
7          [cat = stem, e = ?everb, trans = ?Trans, valnc =
               ?Valnc,
8          valtype = ?Valtype, caus = ?Caus];
9      ?VSTEM -> ?VROOT ; ?VSTEM -> ?CAUSAFF; ?V -> ?VSTEM
10         };
11     <frame> { ?everb[causation,
12                 cause : ?eaff,
13                 effect : ?eroot ]
14     }        }
```

Snippet 16: The class describing causative verbal stems

describes both causative suffixes (i.e., models in which the causative affix follows the verbal root) and causative prefixes (i.e., structures in which the affix precedes the verbal root). Once the test sentence is read, the models leading to incorrect parses are discarded.

The second goal of the class `Causative Verbal Stem` is to describe the semantic frame of the causative verb. Indeed, all categories that can influence verbal semantics are added no higher than the stem level. Inflectional morphemes that adjoin to the stem and form the word of the verb do not contribute to semantics. In line 11 of Snippet 16, the new frame identified as `?everb` (which stands for *event frame of the verb*) is instantiated. It is the frame of type `causation`, which contains two subframes: `cause` and `effect`. The `cause` and the `effect` subframes are not defined in the `Causative Verbal Stem` class but are referenced by their identifiers (variables starting with "`?e`"). Instead, they are described in the classes corresponding to individual morphemes, namely, the causative affix and the verbal root. So, the same structure is used to describe causativized intransitive, transitive, and ditransitive verbs.

The third aim of the class `Causative Verbal Stem` is to describe the operation of valency-increase, i.e., the addition of a new argument into the structure of the verb. Indeed, the valence of the sentence is already conditioned by the verbal stem, in other words, by the verbal root and all derivational verbal affixes. On the semantic side, the increase of valency is done almost implicitly. Each subframe can reference a frame with any number of participants. Thus, the combined frame automatically lists all the arguments required for the whole core and, ultimately, for the whole sentence.

The syntactic part is not as straightforward because, at the level of the verbal stem, no arguments are available yet. For that reason, one needs to account for a specific feature that ensures the correct linking of the required number of arguments higher in the tree. The feature responsible for that is called `valnc`. It is assigned to the node `VSTEM` (see line 7), and its value is imported from the `ValIncrease` class (see line 2). More details about the internal structure of the class `ValIncrease` are given in Section 10.4.3.4. The class `ValIncrease` is imported to the verbal stem along with the classes for morphemes. One could regard it as a rule class that describes an operation rather than its constituents. It relates the transitivity of the verbal root to the valence of the syntactic core and thus tells the expected number of core arguments.

By design, this class is also to be imported into other valency-increasing operations. A similar class would be used for valency-decreasing processes.

The importance of an explicit import of this class can be questioned. The fact that the causative affix increases the valency of the verbal stem is part of the description of the morpheme in the `morpho` dimension (see Section 10.3.1.3.3). One could thus come up with an architecture in which the operation of verbal derivation is somehow incorporated into the syntactic class to which the morphemes anchor. However, I opted for a more modular architecture to better visualize, first, the distinction between constituents and operations, and, second, the different contributions of the verbal stem description to the whole metagrammar. The price for that is the need to explicitly import a rule class. However, since even the most complete metagrammar would not comprise an extremely large number of such classes, I do not consider this slight redundancy a significant disadvantage.

In total, the class `Causative Verbal Stem` generates six models: two ways of ordering the morphemes for each of the three types of frames. The class `Causative Verbal Stem` contains everything needed for a predicate in the causative construction. Therefore, it can easily be united with the `VerbalSpine` (see Section 10.4.3.5). In order to avoid non-lexicalized trees, the class `VerbalSpine` is imported into the class `Causative Verbal Stem` and thus, no parsing operations such as substitution are needed. This approach is parallel to the approach used for nominal classes. Hence, it is the class `Causative Verbal Stem` that gets imported into the class `All Range Causative Constructions` (see Section 10.4.5), which, later, gets intersected with language plugins.

10.4.3.3 Verbal inflection

All verbal inflectional affixes do not influence the analysis of causative constructions. Thus, they are modeled as adjuncts to the `?V` node. Similar to nominal inflectional affixes, all verbal inflectional affixes belong to the same category. The class `Anchor Adjunct Verbal Inflect Affix` creates a single model having a `V*` node that can adjoin to another `V` node dominating the `AFF` node to which morphemes anchor. These trees are added to the trees generated by the other classes listed in this section via sister adjunction during the parsing.

10.4.3.4 Verbal Derivation: valency increase

```
1 class ValIncrease
2 export ?Valtype ?Trans ?Valnc
3 declare ?Valtype ?Trans ?Valnc
4 {?Valtype = valincr;
5     {?Trans = intr; ?Valnc = 2} |
6     {?Trans = tr; ?Valnc = 3}   |
7     {?Trans = ditr; ?Valnc = 4}
8 }
```

Snippet 17: The class modeling the valency increase

The class `ValIncrease` is shown in Snippet 17. This class has no dimension. It operates only with features, which can then appear anywhere in the metagrammar. The purpose of this class is to increase the number of arguments in the CORE of a sentence by adding 1 to the number of arguments of the verbal root.

Since only conjunction and disjunction are allowed, one needs to declare two different variables and use their conjunction to get the respective values. The variable `?Trans` is used in conjunction with the variable `?Valnc`, which stands for *valence*. Thus, each of the values of the variable `?Trans` is paired to a certain value of the variable `?Valnc`. Once a class with a feature `[trans = ?Trans]` gets conjoint with the class `ValIncrease`, all three disjoint options become available. Once the value of the variable `?Trans` is determined, only one option is selected.

```
{?Valtype = valincr; ...
    ?4ArgumentsAllowed = no; ?Trans = ditr; Valnc = 3
...}
```

Snippet 18: An extension of the class `ValIncrease` blocking four-argument cores

Importantly, all these conjunctions are allowed only under the condition listed in line 4: `?Valtype = valincr`. It ensures that the rule describes only the valency-increase operation. This single class can be used for all valency-increasing operations across all languages, since the arithmetic of adding another argument is universal. However, languages may impose restrictions on the number of arguments a verb can take. In this case, additional clauses prohibiting increases in valency would be introduced under certain conditions. An example blocking 4-argument cores is given in Snippet 18. This clause is conjoint with the (imaginary) language-level boolean feature `?4Arguments Allowed = no` set to negative.

10.4.3.5 Verbal Spine

The class `VerbalSpine` (see Snippet 19) is defined to instantiate all the layers of the sentence structure so that all other classes can operate on them. The verbal spine is the

```
class VerbalSpine
    export ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V ?everb
    declare ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V ?everb
    {  <syn>  {
    node ?SENTENCE [cat = sentence];
    node ?CLAUSE [cat = clause];
    node ?CORE [cat = core, e = ?everb, valnc = ?Valnc, caus = ?Caus];
    node ?NUC [cat = nuc, e = ?everb, valnc = ?Valnc, caus = ?Caus] ;
    node ?V  [cat = verb, e = ?everb, valnc = ?Valnc, caus = ?Caus];
    ?SENTENCE -> ?CLAUSE; ?CLAUSE -> ?CORE; ?CORE -> ?NUC; ?NUC -> ?V
}     }
```

Snippet 19: The class describing the verbal spine

same for all types of verbs in all languages; therefore, this class has to be compatible with a variety of other classes. As shown in Section 10.4.2.3 for the nominal spine, the best compatibility is achieved when a non-lexicalized spine class is imported into the target class.

The variation of verbs occurs below the ?V node. Therefore, the verbal spine creates nodes for all levels on top of the ?V node, including the ?V node itself, namely: ?SENTENCE, ?CLAUSE, ?CORE, ?NUC, ?V. Once imported into another class, the verbal spine brings all these nodes with it. Thus, the class Causative Verbal Stem only needs to set the dominance constraint ?V -> ?VSTEM telling that the ?VSTEM node defined in the Causative Verbal Stem class is the daughter of the ?V node imported from the VerbalSpine class. Thanks to that, the entire range of layers becomes available to support parsing operations for sister adjunction and wrapping.

Since the arguments of a construction are filled within the CORE, the features valnc and caus are relevant for all layers below it (including the CORE level itself). In the class VerbalSpine, they are allowed to percolate up to desired levels.

The derived tree for a sample verb is shown in Fig. 40. The class VerbalSpine is combined with the class Causative Verbal Stem and produces a single tree rooted in SENTENCE and anchored in VROOT. The class Causative Verbal Stem has a substitution node for the causative affix, which comes from a different elementary tree at the parsing stage. The inflectional morpheme anchors to a separate tree that attaches to the main tree via sister adjunction.

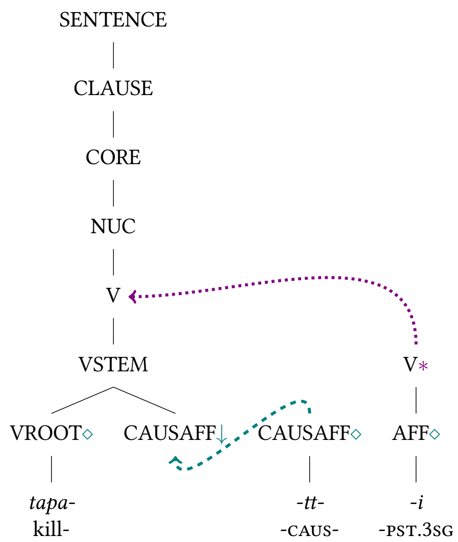


Fig. 40. The derivation of the tree for the Finnish verb *tapa-tt-i*

10.4.4 Word order rules

As one might have noticed, no constraints on word order have been introduced so far. This has been done following linguistic evidence. In most languages, word order is not a device for building causative constructions. For sure, in each sentence, all the constituents are linearly organized. However, most argument realization patterns do not enforce any word order (although, of course, not all theoretically possible combinations are equally frequent). If the goal of the metagrammar were to generate correct sentences in each language, one would need to introduce many constraints to prevent impossible or improbable sentences. While it is not the case, sentence parsing can be done without additional rules for word order.

The situation is different with constructions where word order plays a role in shaping the argument realization pattern. This is relevant for constructions with morphological doubling, in which two constituents appear identical. Usually, in these situations, languages enforce specific word orders to resolve morphological ambiguity.

To parse both kinds of causative constructions – those with a fixed word order and those with a (relatively) free word order – most efficiently, one first needs to distinguish them from each other. The general idea is to create a separate XMG class for each word order. These classes have to be imported into construction classes that get intersected with language plugins (see Section 10.4.5.1, Section 10.4.5.2) to add precedence constraints to the nodes of the final tree. To enable these rules for some languages only, one needs to add a specific feature to the language plugins and, at the same time, set it to a positive value in the rule class. With this approach, the positive value of this feature in the language plugin gets unified with the positive value in the rule; hence, only models

```
1  class ObjectBeforeVerb
2  export  ?LanguageFeatures ?NP_Causer ?NP_Causee
       ?NP_Theme ?CaseCausee ?CaseTheme ?MrCausee ?MrTheme
       ?F2 ?NUC
3  declare ?LanguageFeatures ?NP_Causer ?NP_Causee
       ?NP_Theme ?CaseCausee ?CaseTheme ?MrCausee ?MrTheme
       ?F2 ?NUC
4  { <syn>{
5       {?LanguageFeatures.objectbeforeverb = yes;
6       node ?NP_Theme [case =
           ?LanguageFeatures.MorphUgCase];
7       node ?NUC;
8       ?NP_Theme >> ?NUC}
9       | {?LanguageFeatures.objectbeforeverb = no}
10 } }
```

Snippet 20: The class describing the placement of the object immediately before the verb

with the required word order get generated for each of these languages. If no unification takes place, the word order rule does not apply.

An example of such word order rules is the class `ObjectBeforeVerb`. It has been developed to account for constructions in Turkic and broader, Altaic languages with an unmarked accusative. According to Kamali (2015) citing Öztürk (2005), unmarked accusatives have to be placed in immediate precedence of the verb. This small bit of information constitutes the essence of the class `ObjectBeforeVerb` shown in Snippet 20. It contains two disjoint options. Either the value of the feature `objectbeforeverb` of the variable `?LanguageFeatures` (stored in the Language Plugin) is negative (line 9) – in this situation, nothing happens. Or this value is positive (line 5), in which case the node `?NP_Affectee` is placed right in front of the node `?NUC` containing the verb (line 8). Note that all these variables must be declared in this class and exported from it. This is needed to make all these variables, as well as the features, accessible to other classes that import the class `ObjectBeforeVerb`. The declaration and export of all variables might seem verbose, but they lead to an effective metagrammar compilation.

The question of what to include in a word-order rule is open to discussion. The above-mentioned Turkic causative constructions also have a very strong tendency towards placing the causer clause initially. However, this trend is not part of the causative construction itself. Other PSAs, especially if they are agents, tend to appear in the same syntactic position as well. Thus, this trend could be described by a rule called `PSA Initial` that would be applied not only to causative but to all other constructions.

10.4.5 Constructions

This section finally presents classes that describe whole constructions, i.e., sentences with a predicate and three arguments. There are different classes for dependent-marking

```
class AllRangeCausConstr
import CausativeVerbalStem[]
export ?CORE ?LanguageFeatures ?NP_Causer ?NP_Causee
    ?NP_Affectee ?NP_Recipient ?CaseCausee ?CaseAffectee
declare ?LanguageFeatures ?NP_Causer ?NP_Causee
    ?NP_Affectee ?NP_Recipient ?CaseCausee ?CaseAffectee
{?Caus = yes;
<syn> {
     node ?NP_Causer (mark = subst)
       [cat = np, case = ?LanguageFeatures.MonphPsaCase,
       mr = actor, e = ?F0];
     ?CORE -> ?NP_Causer;
{    { ?Valnc = 2;
     node ?NP_Causee (mark = subst)
       [cat = np, case = ?LanguageFeatures.MonphUgCase,
       mr = ug, e = ?F1];
     ?CORE -> ?NP_Causee
   } |
     { ?Valnc = 3;
     node ?NP_Causee (mark = subst)
       [cat = np, case = ?CauseeCase, e = ?F1, mr = nmr];
     node ?NP_Affectee (mark = subst)
       [cat = np, case = ?AffecteeCase e = ?F2, mr = ug];
     ?CORE -> ?NP_Causee;
     ?CORE -> ?NP_Affectee
}  }  }  }
```

Snippet 21: The class describing all possible causative templates

sentences (see Section 10.4.5.1) and for head-marking languages (see Section 10.4.5.2).

Both classes are organized similarly. Each of them imports the class `Causative Verbal Stem` and creates substitution nodes for arguments. What is also common for both these classes is that they can account for all types of causative constructions, whatever the valency of their base verb. Since the focus of the present work is on causatives-of-transtivies, those of intransitives are only briefly sketched, and those of ditransitives are outlined but commented out to save computational resources. Nevertheless, once the metagrammar is expanded to describe these phenomena in more detail, no new classes would need to be created, significantly reducing the time required to cover new phenomena.

10.4.5.1 Dependent-marking languages

10.4.5.1.1 Generating all possible causative constructions The class `All Range Causative Constructions`, see Snippet 21, is the product of all previously defined classes and, in theory, generates all possible argument patterns from all types of base verbs. (In reality, causatives-of-ditransitives are not fully supported yet since they are outside of this project's focus; for the same reason, they are not displayed in Snippet 21.) This class specifies the features and their values that are universal (or at least hold across all languages in my sample), without specifying the aspects in which causative constructions vary.

The class `All Range Caus Constr` opens with the constraint `?Caus = yes`, (line 5 in Snippet 21), which limits its usage. Thus, other constructions involving a predicate and three arguments would not be described by this class, even though they might have the same tree shape. The variable `?Caus` is used as a placeholder value for the attribute `caus`, standing for *is causative?* in all verb-related classes, starting from the causative affix and higher. So, its assertion as positive checks that the tree contains a causative affix, whose positive value of the attribute `caus` comes from the Lexicon. In case the grammar gets expanded to cover lexical causatives, this constraint will still check whether there is a morpheme with causative meaning, but in contrast to morphological causatives, the bearer of this feature will be the verbal root, not a special affix.

The syntactic dimension opens with a constraint that holds for all causative constructions (lines 7-9 in Snippet 21). It tells that in the tree, there is an NP that bears the PSA case and the actor macrorole. Line 10 tells that this NP is placed below the CORE node. The name of the variable tells us that this NP denotes the causer. However, variable names do not play any role in the sentence representation, so variable names serve mainly for the reader's convenience. The connection between the NP in question and the causer is ensured by the feature `[e = ?F0]`. The `e` attribute relates the syntactic node to the frame, and the `?F0` variable stands for the agent-like participant of the `effect` event in the frame.

The rest of the code is realized as a disjunction between the bivalent cores (i.e., causatives-of-intransitives) and trivalent cores (i.e., causatives-of-transitives). As already mentioned, cores of valency 4 derived from ditransitive base verbs are excluded, as they generate many models, all of which are outside the scope of the current project.

The part describing trivalent cores introduces two additional NPs, since two additional participants are required for these constructions. All of the NPs are substitution nodes. The NP nodes are linked to the semantic structure in the same way as the causer NP: via the feature `e`. The cases to these NPs are assigned in most general terms using variables `?CauseeCase` and `?AffecteeCase`. This class can not specify which cases are used to perform these functions, as no language information has been provided yet. No ordering constraints are introduced for the same reason, as the class `All Range Caus Constr` captures generalizations found in all languages, independently of their structure, including the word order.

As Part II has shown, there is a variety of causee marking in three-argument causative constructions. One option would be to offer all these specifications here, but they would have overcrowded the class. In line with the general principle of modularity, I decided to separate these two types of information into distinct classes.

```
class CausOfTrans
import AllRangeCausConstr[] ObjectBeforeVerb[]
export ?CORE ?LanguageFeatures
declare ?LanguageFeatures
{ ?Trans = tr;
<syn> { ?AffecteeCase = ?LanguageFeatures.MorphUgCase;
{
    {?LanguageFeatures.BoolCauseeLikeRecip = yes;
    ?CauseeCase = ?LanguageFeatures.MorphRecipCase
} |
    {?LanguageFeatures.BoolCauseeLikePassDemAg = yes;
    ?CauseeCase = ?LanguageFeatures.MorphPassDemAgCase
} |
    {?LanguageFeatures.BoolCauseeLikeCvbAg = yes;
    ?CauseeCase = ?LanguageFeatures.MorphCvbAgCase
} |
    {?LanguageFeatures.BoolCauseeLikeInstrument = yes;
    ?CauseeCase = ?LanguageFeatures.MorphInstrCase
} |
    {?LanguageFeatures.BoolCauseeLikeUg = yes;
    ?CauseeCase = ?LanguageFeatures.MorphUgCase
}  }  }  }
```

Snippet 22: The class specifying argument coding in causatives of transitives

10.4.5.1.2 Specifying the argument encoding A separate class entitled `Caus Of Trans` (which obviously stands for *causatives of transitives*) has been created to specify the argument encoding in the target constructions. This class is presented in Snippet 22 and is notable for its clarity and effective organization, as it includes highly specific information.

The class opens with a constraint `?Trans = tr`, which tells that only causatives derived from transitive base verbs are relevant for this class. As the class `All Range Caus Constr` creates tree shapes for cores of all valency and derivations of all types of verbs, this constraint filters out irrelevant models.

Currently, the class `ValIncrease` (see Snippet 17, Section 10.4.3.4) creates three-argument causatives from transitive bases only. Thus, in the current metagrammar, the constraints `?Trans = tr` and `?Valnc = 3` are absolute synonyms. However, the real situation might be more complex. Some languages, e.g., Trumai (Section 6.2.2.3) and Kabardian (Section 6.2.2.2), have the so-called extended intransitive verbs. The causative constructions derived from extended intransitives also have three arguments but use a different argument-marking strategy. If those verbs were accounted for in the `class ValIncrease` (e.g., by adding a constraint `?Trans = extintr; ?Valnc =3`), there would be another type of constructions with `?Valnc = 3`, which would not be derived from transitive verbs. These constructions would require a separate class specifying their argument encoding. Although the current metagrammar does not describe causatives of extended intransitives in detail, I keep that in mind as a possible expansion. Therefore, in the class CausOfTrans, I use a more precise constraint ?Trans = tr, even though it currently yields identical results as a potentially wider constraint ?Valnc = 3.

In the syntactic part, no additional nodes are declared since all of them have already been defined in the class All Range Caus Constr. What is left for the class CausOfTrans, is the specification of the variables ?Causee Case and ?Affectee Case, which were used as values for the feature [case] in the class All Range Caus Constr. Thus, the variables ?CauseeCase and ?AffecteeCase are a middle link in the chain uniting the class All Range Caus Constr and the language plugins. In the class CausOfTrans (see Snippet 22), these variables are assigned the same values as the case features in language plugins. For accessing them, the dot notation (see details in Section 9.5) is used. This access becomes critical once the class CausOfTrans gets intersected with a language plugin (see Section 10.4.6).

In line 6 of the class CausOfTrans (Snippet 22), the case for the affectee is specified as the case used for prototypical undergoers in a simple monotransitive construction. This part falls outside all disjunctions, since the affectee bears the same functional case in all the languages in my sample. In contrast, the causee case depends on the strategy used by a language, yielding multiple options listed in lines 8-21.

The remainder of the class CausOfTrans (Snippet 22) is organized as a multiple disjunction of conjunctions, which can read as the following implication: if the boolean is true, the given case is assigned. The booleans tell whether a certain type of construction is attested in a language or not (see more details in Section 10.5). Naturally, all these boolean features are accessed via the variable ?LanguageFeatures because their values vary by language.

The true value of a boolean triggers the assignment of a specific case to the variable ?CauseeCase, which, in turn, transmits this value to the node ?NP_Causee defined in the class All Range Caus Constr. The values of the cases are also language-specific and retrieved via the dot notation from the ?LanguageFeatures structure.

Each pair of constraints relates to one argument-marking strategy. It does not tell anything about the co-occurrence of these strategies in a single language. If a language plugin has only one boolean set to positive, only one pair of constraints of the class CausOfTrans has effect, and, subsequently, only one tree is considered correct. If a language has several causative constructions, its plugin has several boolean variables with positive values, which, in turn, trigger several pairs of constraints of the class CausOfTrans.

The class CausOfTrans is also the place where word order rules are applied. Namely, in line 2 of Snippet 22, the class ObjectBeforeVerb is imported. Since the class CausOfTrans gets intersected with language plugins, everything imported into it also benefits from this intersection. Thus, the word-order rule applies or not depending on

```
1  class CausOfTransHM
2  import CausativeVerbalStem[] ObjectBeforeVerb[]
3  export ?V ?Caus ?Valnc ?ARGAFF_Causer ?ARGAFF_Causee
       ?ARGAFF_Affectee ?NUC
4  declare ?V ?ARGAFF_Causer ?ARGAFF_Causee
       ?ARGAFF_Affectee ?NUC
5  { <syn> {
6  { ?Valnc = 2;
7    node ?ARGAFF_Causee (mark=subst) [cat = argaff];
8    node ?ARGAFF_Causer (mark=subst) [cat = argaff];
9        ?V -> ?ARGAFF1; ?V -> ?ARGAFF2
10     }
11     |
12 { ?Valnc = 3;
13   node ?ARGAFF_Affectee (mark=subst)
14     [cat = argaff, case =
          ?LanguageFeatures.MorphUgCase, e = ?F2, mr = ug];
15   node ?ARGAFF_Causee (mark=subst)
16     [cat = argaff, case =
          ?LanguageFeatures.MorphRecipCase, e = ?F1, mr =
          nmr];
17   node ?ARGAFF_Causer (mark=subst)
18     [cat = argaff, case =
          ?LanguageFeatures.MorphAgCase, e = ?F0, mr =
          actor];
19   ?V -> ?ARGAFF1; ?V -> ?ARGAFF2; ?V -> ?ARGAFF3
20 }  }  }
```

Snippet 23: The class describing causative templates in head-marking languages

the language plugin's specifications.

10.4.5.2 Head-marking languages

Although I have tried to include head-marking languages in my sample, they are still significantly outnumbered by dependent-marking ones. All of them have the same construction **Causative like Ditransitive** with the single argument-marking strategy **Causee like Recipient**. Therefore, there is no need to create two separate classes, as in dependent-marking languages (see above). The initialization of the tree shape as well as the specification of the cases, are done in the single class `CausOfTransHM` (see Snippet 23). Nevertheless, if the metagrammar is updated to cover more head-marking languages and they use different argument-marking strategies, a proper separation of the two types of information will be required.

Like the class `All Range Caus Constr` for dependent-marking languages, the class `CausOfTransHM` builds causatives of all types of verbs of any valence. Once again, the bivalent causatives (where `?Valnc = 2`) are only briefly outlined, and causatives where `?Valnc = 4` are commented out.

The main purpose of the class `CausOfTransHM` is to combine the verbal stem with argument affixes. For that, the class `Causative Verbal Stem` is imported (see line

2 in Snippet 23). It already imports the class `VerbalSpine`, so all the layers of the clause above the `VSTEM` level are available for the class `CausOfTransHM` as well. Thus, for building trivalent causatives, the class `CausOfTransHM` creates three `?ARGAFF` nodes (standing for *argument affixes*) that are dominated by the `?V` (verb as a word) node, see line 19 in Snippet 23. All of these nodes are substitution nodes taking elements of category `argaff`. These elements are anchored in the class `ArgumentHM`, see details in Section 10.4.2.4. The arguments are connected to frames via the `e` feature.

Since there are three argument affixes, all of which are direct core arguments, each can be assigned a macrorole status. This is done using the feature `mr`. Note that macroroles are related to semantics: causers are always actors, affectees are always undergoers, and causees are always NMR participants. The existence of these three affixes is universal.

In contrast, the order of the affixes can vary depending on the language, as well as the position of the affixes in relation to the verbal stem (prefixes or suffixes). As one can see in Snippet 23, no such constraint is added to the class.

Since, in my sample, all head-marking languages demonstrate the same argument-marking strategy in causatives-of-transitives, the case specifications are added directly into the class `CausOfTransHM`. Since this class gets intersected with language plugins, all case values are accessed via the `?LanguageFeatures` variable using the dot notation. For the same reason, the class `ObjectBeforeVerb` is imported into the class `CausOfTransHM`.

10.4.6 Intersections

```
class IntersectionFIN
import CausOfTrans[] Plugin_FIN[]

class ArgumentFIN
import NounWithCase[] Plugin_FIN[]

class IntersectionADY
import CausOfTransHM[] Plugin_ADY[]

class ArgumentADY
import ArgumentHM[] Plugin_ADY[]
```

Snippet 24: Intersection classes (exemplified by Finnish and Adyghe)

Intersections are the final classes for nominal and verbal root morphemes, in other words, the highest classes of the whole metagrammar. These classes are set as families for lexical morphemes, ensuring correct parsing of the sentences. The key point of the provided solution is that these classes have an extremely simple architecture, leaving the filtering of the required models to other classes.

For each language, there are two intersection classes (see Snippet 24 for examples). None of them has any dimension open, no new nodes are created, and no feature values are assigned. The only thing that is done in intersection classes is `import`. Each intersection class imports one construction class and one language plugin. As demonstrated in Snippet 24, the final class for nominal lexemes is the class `ArgumentLAN`, where `LAN` stands for the language code. It imports the construction class `Noun With Case` for dependent-marking languages and the class `ArgumentHM` for head-marking languages. Similarly, the `IntersentionLAN` classes import the classes anchored in verbal roots. These classes are described right above, see Section 10.4.5.1 and Section 10.4.5.2.

From the technical side, what I call an intersection is merely a conjunction. The information from both classes is combined, resulting in unified feature values. Some unifications determine the selection of one of several disjoint options; this is how the filtering of the models is realized. As a result, all specifications for open feature values are provided by the plugin, so that only trees allowed in a given language become available. At the parsing stage, only the elementary trees corresponding to the values coming from the test sentence remain to provide the correct analysis of a sentence.

The intersection classes can be generated automatically using a script, thus reducing development time and eliminating possible human errors. Importantly, after having been created, each of these classes has to be valuated to make the models visible to the parser.

10.5 Language Plugins

10.5.1 Description

From the linguistic (typological) point of view, as well as in line with RRG's view on constructions, individual languages do not add anything new to the general definition of a grammatical phenomenon. Instead, they are supposed to be variants, possible realizations of a theoretical abstraction. It seems logical that, in terms of grammar engineering, language-specific information must not be encoded as nodes but rather as features assigned to the nodes of the abstract generalization.

In line with this logic and the place of constructional schemas in RRG (Section 2.7), I created a special type of XMG classes called `Language Plugins`. In the solution presented here, each class encodes all the information necessary for parsing three-argument causative constructions in one language. In a larger project, it would encode more grammatical features (ideally the full grammar of a language), similarly to the questionnaire in the LinGO Grammar Matrix (Bender, Drellishak, et al. 2010).

Each language plugin (see example in Snippet 25) comprises one variable called `?Language Features` (line 4), which is a very large feature structure. Note that

```
class Plugin_BAK      %% Bashkir
  export ?LanguageFeatures
  declare ?LanguageFeatures

  {?LanguageFeatures = [
    objectbeforeverb = yes,
    locusofmarking = dep,

  %% Inventory booleans
    BoolCauseeLikeRecip = yes,
    BoolCauseeLikePassDemAg = yes,
    BoolCauseeLikeCvbAg = no,
    BoolCauseeLikeInstrument = no,

  %% Morphology enabled by true booleans
    MorphPsaCase = nom,
    MorphUgCase = acc,
    MorphPassDemAgCase = abl,
    MorphRecipCase = dat,
    UnmarkedCases = @{nom, acc}
  ];
  <iface>{[_lang_ = BAK]}}
```

Snippet 25: The language plugin for Bashkir

this variable is declared outside of any dimension.

Once language-specific information has to be called in the metagrammar, the respective class declares a variable of the same name `?LanguageFeatures` and references the required feature via the dot notation (see Line 6 in Snippet 26). When a construction class intersects with a language plugin, the variables `?LanguageFeatures` unify, and the values of the features defined in the plugin percolate to the construction.

Snippet 26 demonstrates a fragment of the Construction class `All Range Caus Constr` followed by fragments of two Language Plugins. The node `?NP_Causer` has a feature `case` assigned. Its value is not hard-coded (explicitly written) but is referenced via dot notation. Line 6 tells that the value of the feature `case` of the node `?NP_Causer` has to be the same as the value of the feature `MorphPsaCase` on the variable `?LanguageFeatures`. It is important to highlight that the variable `?LanguageFeatures` has to be declared in this class (which is done in line 4); otherwise, the reference via dot notation will not work. The feature `MorphPsaCase`, however, does not necessarily have to be declared in this class. The fact that this feature exists is reflected in the preamble of the metagrammar so that it can be referred to. However, when the value of this feature is not specified (for instance, when one compiles the class `All Range Caus Constr` without intersecting it with any language plugin), it is realized just as a placeholder. Once the intersection occurs, the value is determined. Thus, when intersected with `Plugin_KBD`, the node `?NP_Causer` gets `case = erg`. Respectively, when intersected with `Plugin_KAN`, the same node gets `case`

```
1  class AllRangeCausConstruction
2  import VerbalSpine[]
3  export ?CORE ?LanguageFeatures ?NP_Causer ...
4  declare ?LanguageFeatures ?NP_Causer ...
5  {?Caus = yes;
6  <syn>{node ?NP_Causer (mark = subst) [cat = np, case =
       ?LanguageFeatures.MorphPsaCase, mr = actor, i =
       ?F0];...}}

8  class Plugin_KBD  %% Kabardian
9    export ?LanguageFeatures
10   declare ?LanguageFeatures
11   {?LanguageFeatures = [MorphPsaCase = erg, ...]}

13 class Plugin_KAN  %% Kannada
14   export ?LanguageFeatures
15   declare ?LanguageFeatures
16   {?LanguageFeatures = [MorphPsaCase = nom, ...]}
```

Snippet 26: The usage of `?LanguageFeatures` in Construction Classes and Language Plugins

= `nom`. As lines 11 and 16 show, the variable `?LanguageFeatures` is declared in language plugins outside of any dimension (cf. its appearance in `<syn>` dimension in line 6). This declaration makes it possible to reference the `?LanguageFeatures` variable anywhere in the metagrammar and get it unified with the very same instance of the language plugin.

The internal structure of language plugins is almost identical, comprising all the features that are necessary to parse target sentences in each language. Thus, for the sake of economy, only those cases that are relevant for parsing causative constructions are listed in each plugin. The features that are assigned to the `?LanguageFeatures` variable are of different types. Each one is listed in the preamble of the metagrammar and assigned the respective type. In other words, the metagrammar checks that every feature listed in every class is declared to exist and takes values of the allowed type.

One linguistic feature already demonstrated in this section is morphological features that define nominal cases. They take the type `CASE`, which in turn is defined as a closed set of values, namely, case abbreviations (see respective lines of the preamble in Snippet 27).

```
type CASE = {abl, abs, acc, adess, all, dat, dat-acc,
    det, dir, erg, ins, nom, obl, par}
feature MorphPsaCase : CASE
feature MorphUgCase : CASE
feature MorphRecipCase : CASE
feature UnmarkedCases : CASE
...
```

Snippet 27: Case features defined in the preamble

The cases are defined in functional terms. In other words, the names of the features tell in what syntactic function or for what semantic role the morphological case is prototypically used. The sole exception is the feature `UnmarkedCases` (see details in the next section).

Another type of feature in the language plugins is the so-called **inventory booleans**. As the name suggests, these features take one of the two values: `yes` or `no`. The term *inventory* is used in the sense that these features tell what variants of causative constructions (what inventory) are available in a given language. Thus, Snippet 28 tells that in Bashkir, two causative constructions are possible: the one where the causee is realized as the recipient and the other where the causee is realized as the demoted agent.

```
class Plugin_BAK ...
  {?LanguageFeatures = [
    BoolCauseeLikeRecip = yes,
    BoolCauseeLikePassDemAg = yes,
    BoolCauseeLikeInstrument = no,    ...]}
```

Snippet 28: Inventory booleans in Bashkir

The list of these features is the same in all language plugins. Thus, it is explicitly stated about each construction whether it exists in a language or not. If the value of a feature remains undetermined, no filtering occurs, and all technically possible models are generated.

The third type of features present in the language plugin can be called *language-level features*. They describe some general characteristics of a language that are relevant for parsing any type of construction and, in particular, causatives. These features are listed in lines 6 and 7 in Snippet 25. The first one, `object before verb`, is responsible for the application of the word order rule to a given language (as described in Section 10.4.4). The second one, `locus of marking`, selects the responsible families of trees.

10.5.2 Unmarked cases

In many languages, there are morphological cases that have no explicit marker. These situations are a real challenge for the metagrammar and the parser. In glossed linguistic examples, one usually introduces a special character to show that there is no overt morpheme, but the noun does have a case (see (126), identical to (21)).

(126)	nēnu	rāmu-cēta	kāru-∅	kaḍig-inc-ǣ-nu	Telugu
	I	Ramu-ins	car-acc	wash-caus-pst-1sg	

‘I had the car washed by Ramu’/ ‘I caused the car to be washed by Ramu.’

Krishnamurti 2003, p. 433, (20f)

When designing a metagrammar, one could have adopted one of the two approaches in representing nouns like “car” in (126). The first line would follow the notation of linguistic examples. Thus, one would add the morpheme "∅" into the parsing input, which is not the actual sentence but rather a string of morphemes. Subsequently, in the lexicon for the respective language, one would need to introduce a special morphological entry, see Snippet 29.

```
class CaseMarker_0
import Abstr_CaseMarker[]
{<morpho>{
morph <- "∅";
case <- nom;
lemma <- "Nom" }}
```

Snippet 29: The `morpho` class for a case marker realized as morphological zero

In fact, adding a special symbol to the input line could be considered a minor adjustment to the example. However, this solution is not the best possible for several other reasons. The most crucial drawback is the fact that a language can have more than one category that can be realized as morphological zero. To handle all of them, one would need to introduce quite many morphemes with a null `morph`, but different `lemma` and feature structure. That would have led to an increase of ambiguities and overgeneration of trees. To avoid that, one would need to design, introduce, and apply additional restrictions on feature and morpheme combination, which seems to be a lot of work of questionable necessity. Second, if one decides to introduce special characters for null morphemes, one needs to be consistent in this approach. Otherwise, the modified example would not make linguistic sense. An attentive reader might have already noticed that (21) contains another word without an overt case marking: "nēnu". However, in the source paper (Krishnamurti 2003, 433), no case has been assigned to this pronoun. Neither the "∅" morpheme is added to the gloss line. (Nevertheless, pronouns do have case in Telugu, and "nēnu" is nominative, see Krishnamurti (2003, 245).)

Since similar situations are very frequent in languages, a less error-prone and more consistent way to handle unmarked cases would require no additional entries to the grammar lexicon. Given these complications, the second way of representing unmarked cases seems more promising. The idea behind it is to represent morphological case as a feature that does not necessarily require a segmental node. In other words, `case` is a feature assigned to the NP, but it does not mean that it necessarily has to be assigned to anything else. The syntactic details of this decision are given in Section 10.4.2.1. The section below focuses on the representation of the unmarked cases in language plugins.

A plugin has to tell what cases can be unmarked in a given language. Note that quite often, there are two cases in a language that can be realized as null morphemes. Thus, it is insufficient to introduce one feature `[UnmarkedCase : CASE]` (like other case-related features described in Snippet 27) because it cannot tell if two cases can be unmarked. An alternative approach would be to introduce boolean features for each

```
class Plugin_TEL ...
  {?LanguageFeatures = [
    UnmarkedCases = @{nom, acc}
  ] ... }
```

Snippet 30: Unmarked cases in Telugu (fragment of a language plugin)

case, indicating whether it can be unmarked or not, e.g., `PsaCaseUnmarked = yes`, `RecipCaseUnmarked = no`, etc. However, too many features could overload the metagrammar, and their repetitive introduction could lead to errors.

To offer an effective and economical solution, I resorted to a special data type of XMG called "atomic disjunction of constant values" (see details in Section 9.4.3) I defined a feature `UnmarkedCases` of type `CASE`, but instead of assigning it a single value, I offer a disjunction of several cases. The disjoint values are written in curly brackets preceded by the "`@`" symbol, see Snippet 30. To highlight: all values within this structure have to belong to the declared type (`CASE` for instance).

```
1  class Plugin
2    export ?LanguageFeatures
3    declare ?LanguageFeatures
4    {?LanguageFeatures = [
5      MorphPsaCase = nom,
6      MorphUgCase = acc,
7      MorphRecipCase = dat,
8      UnmarkedCases = @{nom, acc}
9    ] }

11 node ?NP [cat = np, case =
       ?LanguageFeatures.UnmarkedCases, mr = ?Mr]

13 node ?NP-1 (mark = subst) [cat = np, case =
       ?LanguageFeatures.MorphPsaCase, mr = actor]

15 node ?NP-2 (mark = subst) [cat = np, case =
       ?LanguageFeatures.MorphUgCase, mr = ug]

17 node ?NP-3 (mark = subst) [cat = np, case =
       ?LanguageFeatures.MorphRecipCase, mr = nmr]
```

Snippet 31: A language plugin in action: some examples

The value assigned through atomic disjunction remains underspecified until it gets unified with another feature (if any). Snippet 31 demonstrates some code fragments to illustrate this process. One has a `Plugin` for some language, where both NOM and ACC cases can be unmarked, which is stated in line 8. A node `?NP` is declared in line 11, and the value of the feature `UnmarkedCases` from the language plugin is assigned as the value of the feature `case` of this node. From now on, this node can unify with another node whose feature `case` has the value `nom` or `acc`. For instance, when `?NP` is to come as a substitution node below `?NP-1` in line 13, the substitution is effective. As we know from the language plugin, `MorphPsaCase = nom` (line 5), and `nom` is one of the possible values for the feature `case` of the node `NP`. The resulting tree has `case = nom` throughout, with no disjunctions left. Respectively, `?NP` can also substitute `?NP-2` in line 15, setting the `case = acc` for the whole tree. Unlike those, the substitution in line 17 can not happen. As the Plugin tells, `MorphRecipCase = dat` (line 7), so the feature `case` for `?NP-3` has the value `dat`. It is not among the allowed values for `?NP`, because it is not listed in the atomic disjunction set as the value of the feature `UnmarkedCases`. So, the features cannot unify, and the substitution cannot take place.

10.5.3 Grouping language plugins

Once one wants to cover more languages with a metagrammar, one would need to create many language plugins following the same template. Apparently, some features are related to others because of the genetic or typological similarity of the languages. To reduce repetition and avoid errors when filling data, one could create small plugins containing related information to reuse across several classes. This idea is inspired by the abstractions used to create lexical files, as demonstrated in Section 10.3.

However, the idea of grouping language plugins does not make as much sense as it might seem. The main challenge is that linguistic features vary along multiple axes. Selecting the only one to freeze as an abstraction would be technically challenging. One could, though, use some insights from the principal components method to determine the axis of greatest variation. However, the resting variation across other axes would still require additional annotation. Applying all of these approaches would lead to a less intuitive plugin structure, which, in turn, might reduce its interpretability for users and annotators of the metagrammar.

The main concern against grouping language plugins is, in fact, the difference between even closely related languages. As has been determined in Chapter 7, causative constructions demonstrate quite a variation even within language families and groups.

To conclude, it appears that creating abstractions from language plugins is possible after all the individual languages are described. This could, perhaps, highlight some genetic and/or typological similarities. However, it is not particularly helpful for reducing the annotation workload.

10.5.4 Discussion

Language Plugins provide language-specific information in an encapsulated, concise format. In contrast to Kinyon et al. (2006, 20), where classes describing language-specific information were structurally not quite different from those describing generalizations, Language Plugins have multiple distinctions from Construction Classes. Namely, they have no nodes and are compatible with any dimension.

All the features listed in Language Plugins are kept as similar to each other as possible to allow their seamless intersection with Construction Classes. Moreover, all definitions are given in functional terms, grounded in linguistic evidence and descriptive studies. This facilitates the reading and understanding of the metagrammar by various scholars. Also, functional definitions are more in line with RRG as a theory than just language-based, hard-coded values.

The main drawback of the current approach towards creating Language Plugins is that it requires manual work by qualified specialists. To put it more precisely, a trained linguist with an advanced understanding of grammar engineering has to design the list of features to be enumerated in Language Plugins to work with a given grammar phenomenon. Once this work is done, plugins for new languages would follow the same pattern and could be created by language specialists or even by native speakers without linguistic education.

To ease the workload, one could reuse information from previously defined plugins in newer series that describe other linguistic phenomena. Thus, morphological features (case definitions) and perhaps some language-level features defined here in the metagrammar of morphological causatives could be reused in many other metagrammars: starting with those describing other valency-modifying constructions, but also all those where information about morphological cases and/or word order is relevant. However, one would need to define inventory booleans from scratch for each new type of construction. Also, some new language-level constraints might be needed to describe other phenomena. All of this is an undeniable drawback of the solution presented here – as well as of any solution that relies on predefined rules rather than machine learning. In this respect, the architecture of the language plugins presented in this section is similar to the customization system of the LinGO Grammar Matrix (Bender, Drellishak, et al. 2010): the developer must define the questions that can be answered by anyone willing to create a new grammar. Similarly, the developer of the language plugins needs to define the list of features to be included, whereas it is sufficient to specify their values to add other languages.

Lately, there have been initiatives to support the manual creation of linguistic resources by automatically transcribing features from larger databases (e.g., in some works by the ACL Special Interest Group on Typology[44]). No such work has been done yet with XMG, but it could definitely become a valuable contribution to the growth of the family of metagrammars similar to the one presented in this work. The downside of this approach, already remarked by some scholars, is the scarcity of data in these databases. One could never be sure that the information necessary for the metagrammar is avail-

[44] `https://sigtyp.github.io/index.html`, accessed on 01.05.2026

able in, for instance, WALS (Dryer and Haspelmath 2013) or Grambank (Skirgård et al. 2023). Moreover, the formats of encoding similar features can vary even within the same database, making automatic conversion unstable. Therefore, the duty to design specific features for the metagrammar can still not be taken away from the creators. To decide on the necessity of inventory booleans and determine their list, a thorough theoretical analysis presented in Chapter 6 was needed.

Chapter 11

Evaluation and expansion

11.1 Evaluating my solution

The solution I present in this thesis is difficult to evaluate with methods typically used for computational implementations. It is not possible to split the language sample into a training and a test set because, for each language, special lexical files need to be created. Thus, test languages still require some development; hence, sentences in those languages can not be used for parsing right away.

The same complication arises when one splits the set of sentences available in each language into two parts – training and testing. The parsing of the test sentences would be impossible without the addition of the previously unseen morphemes to the lexical files.

Thus, traditional coverage tests, even to an extent applicable to the presented solution, are not a reliable method for testing it. A different evaluation, perhaps, could bring more insights. One might wish to test the solution using *precision* and *recall*. In this setup, the *recall* would measure the percentage of target sentences parsed correctly by the suggested solution. To assess the *precision*, one would need to determine what counts as a false positive. Since the goal of the suggested solution is to offer syntactic analyses of real sentences in real languages, a *false positive* would be an analysis of an incorrect sentence. Such analyses are produced by my solution mainly because of the morphemes realized as a morphological zero. The results of this testing protocol are suggested in Appendix C.

A thorough testing setup would also require running tests on negative examples and assessing the number of true and false negatives. To test a solution aimed at parsing real sentences, one would need to create a special sample of negative examples. The metagrammatical representation and the parsing of individual words are best performed with unit testing (see below); therefore, the negative examples are full constructions that are, for some reason, ungrammatical.

The first reason is missing words. The present solution aims at parsing only sentences with one verb and three nominals (nouns and pronouns). Thus, causatives of transitives with two nominals instead of three, like the modified example (127), count as ungrammatical and are not parsed by the presented solution. I am aware of the fact that these sentences are often found in real language data because the causee can often be omitted for pragmatic reasons. Handling the optional omission of the causee and prohibiting the omission of the causer and the affectee can become one of the next developments of the project.

(127) FINNISH

Satu	tapa-tt-i	etana-n.
Satu.NOM	kill-CAUS-PST.3SG	slug-ACC

'Satu had (Diane) kill the slug.'

VG: the word *Diane-lla* removed from the original example (55a)

The next known limitation of the presented solution is the presence of extra words, namely, of adjectives modifying nouns or of adverbs. To handle sentences like (128), adjective phrases need to be described in the metagrammar and allowed to adjoin to NPs.

(128) YAQUI

Aurelia-∅	Karmen-ta	u-ka	ili	usi-ta	bit-tua-k
Aurelia-NOM	Carmen-ACC	the-ACC	little	child-ACC	see-CAUS-PSTPFV

'Aurelia showed Carmen the child'. Guerrero and Van Valin 2004, p. 36, (32b)

Another source of the sentence ungrammaticality can be the usage of the wrong morpheme. The parser provides an output tree if all the morphemes in the sentence are determined by the corresponding `Intersection` class, where a construction class and a language plugin merge. Thus, to test this parameter, I have created ungrammatical sentences with unacceptable morpheme combinations. Consider (129) with the word *Diane-n* 'Diane-ACC' insttead of *Diane-lla* 'Diane-ADESS'. This sentence is not parsed because the boolean feature `CauseeLikeUg` is set to negative in the language plugin for Finnish, hence, two ACC arguments are not allowed.

(129) FINNISH

Satu	tapa-tt-i	etana-n	Diane-n
Satu.NOM	kill-CAUS-PST.3SG	slug-ACC	Diane-ACC

'Satu had (Diane) kill the slug.'

VG: the word *Diane* bears an ACC suffix instead of the ADESS

Another dimension of testing is *unit testing*. It suggests testing each block of a complex system separately, which is especially helpful in modular projects like metagrammars. Smaller blocks have less potential for variation; hence, the creation of negative examples is more straightforward. Thus, after designing each class, I was able to check for all the details: node positioning, feature assignment, etc. Unit testing is available thanks to the *axiom* functionality in XMG. The parsing can be performed on structures of any level if the correct *axiom* (the syntactic category of the constituent to be parsed) is given to the parser. For instance, it is possible to check the adjunction of inflectional morphemes to nouns without compiling the whole part of the metagrammar related to verbs and without providing the parser with the whole sentence containing a verb and all the arguments. Instead, a parsing command like (130) can be executed to provide the parsing on a single Finnish noun *naapuri-lla-an* 'neighbour-ADESS-POSS.3'.

(130)
```
java -jar  /TuLiPA-frames/TuLiPA-frames.jar -rrg -g
MainMetagrammar.xml -m Morpho/Auto_Morpho_FIN.xml
-l Lemma/Auto_Lemma_FIN.xml
-a noun -s "naapuri lla an"
-xg -o Output/"FIN-noun1.xml"
```

I am not providing the unit testing protocols, as the results of this procedure have influenced and even determined the final architecture of the solution. Generally, I have tested the following parameters:

1. all morphemes are written correctly,
2. all obligatory morphemes are not missing,
3. optional morphemes can attach,
4. only morphemes of the allowed category do attach,
5. the word is parsed using the lexical files of the same language.

An additional axis of evaluation would be the degree of factorization and the expansion potential of the solution. However, the quantitative measure thereof is once again difficult. Throughout Chapter 10, I have shown that classes describing other parts of speech can be added easily. The same holds for other types of syntactic constituents. Moreover, the same solution can be used for describing lexical causatives. The single change required is the optionality of the causative affix. I believe that this case illustrates well the potential of the presented solution.

11.2 Reproducing the results

The solution presented in this chapter is available as a public GitHub repository: https://github.com/fleurdhelice/Metagrammar_CausativesOfTransitives. It contains all the necessary files to reproduce my results, along with some instructions for modifying the solution (if desired). The file **README.md** contains instructions for using the provided solution, which mainly repeat the contents of this section.

In addition to the **README.md** file, the root directory of the repository contains some additional files and folders. The first file is **MainMetagrammar.mg**. It contains the main metagrammar described throughout this chapter. For quick access, it is also provided in Appendix A. This file has to be compiled (see next section) in order to parse any sentence in any language. The repository already contains a compiled file **Main-Metagrammar.xml** (mind the extension) that can be used for one's own purposes right away. However, if one makes any modifications to the metagrammar (including commenting out intersections for some languages), one needs to recompile the `.xml` file.

Another file is **ParsingCommands.txt**. It contains ready-to-use commands for the parser in case one uses a local installation of XMG and TuLiPa or accesses them via a virtual machine. In these commands, all the file paths, axioms, and sentences in the correct form are put together.

Lexical files are stored in the **Lemma** and **Morpho** folders, respectively. There is one Lemma file and one Morpho file for each language. Once again, each folder contains source metagrammars (`.mg` files) together with compiled metagrammars (`.xml` files). In case metagrammars are modified, new `.xml` files need to be compiled.

Another folder in the root directory is **Sentences**. It stores `.txt` files for each language, each containing pre-processed sentences. These sentences can be used in parsing commands in the virtual machine or in a local installation. For convenience, the same folder also contains files with target sentences accompanied by glosses. All sentences listed in this folder are tested to get parsed correctly. If one would like to process their own examples, the new morphemes will need to be added to the lexical files. Finally, the **Output** folder contains the tree representations produced by the parser.

To run the solution, one needs the XMG compilers and the TuLiPa parser. If one would like to run the solution on their own machine, they might wish to either install everything locally or access the XMG compilers and TuLiPa parser via a virtual machine. Instructions for both these options are given in the official XMG documentation[45].

11.3 Comparison to other projects

11.3.1 Early multilingual metagrammars

Several approaches to creating multilingual metagrammars have been suggested by Alexandra Kinyon and her collaborators. Namely, Clément and Kinyon (2003) suggest generating LFG (Lexical Functional Grammars) and TAG (Tree-Adjoining Grammars) for French and English. Even two languages already trigger a "multilingual setup," drawing attention to one of the aspects of grammar development that comes "for free" with the general metagrammatical approach. Since the metagrammatical describes language *descriptions*, not languages, it follows the logical combinatorial rules. Hence, if two phenomena are

[45]`https://github.com/spetitjean/XMG-2/wiki/1:--First-steps`, accessed on 01.05.2026

attested in different languages but are logically related to the same abstraction, the metagrammar can capture this abstraction and the inheritance relation, and compare constructions across languages. Being multilingual is thus a consequence of being abstract. Nonetheless, the works by Alexandra Kinyon and her team are remarkable for deliberately using this asset and developing multilingual metagrammars. All other projects, including the LinGO Grammar Matrix, give the impression that cross-linguistic generalizations are merely shortcuts for developing the grammars of individual languages.

In this light, the paper "The metagrammar goes multilingual: A cross-linguistic look at the V2 phenomenon" (Kinyon et al. 2006) stands out from the whole prior scholarship as having the same aims as my study. Namely, it describes one given construction (the "V2 phenomenon", i. e., the requirement for the finite verb to take the so-called second position in a sentence) across several languages, including five Germanic languages and Kashmiri (yet the discussion in the paper is limited to German and Yiddish only). It already uses an early version of XMG (Crabbé 2005), which brings this work even closer to mine.

The authors suggest two structures that are similar in all target languages: the maximal projection of a verb and the argument realization. These modules are reused across languages. In addition, the authors develop a set of features assigned to the verbal head to constrain its shape and properties. For each language, the authors generate an inventory of head structures, defining the feature values. My solution is much more concise in this respect, since I store language-specific information in separate classes and retrieve the values of those features as needed, which makes a prior listing of possible trees unnecessary. Finally, the unification mechanisms of XMG compile the trees from tree fragments according to the feature values. One can see that sketched broadly, this solution resembles mine very closely. What is, however, different is the overall higher degree of modularity (provided by my use of the more advanced version of the tool, XMG-2). Thanks to that, not only is the mentioned separation of language-specific information done, but also a linguistically salient separation of dimensions. In my solution, dimensions correspond to levels of language structure (morphology, syntax, semantics). In contrast, Kinyon et al. (2006) uses the dimensions that reflect the technical properties of a tree-based grammar rather than the language data.

11.3.2 FunGramKB + Artemis

A quite well-developed alternative computational model of RRG is suggested by Cortés Rodríguez and Mairal-Usón (2016). They claim that their model primarily focuses on NLU (natural language understanding); hence, its primary focus is on ontological semantics. The syntax serves just as an entry point leading to semantics. The authors themselves describe their project as the unidirectional "computational replica of the syntax-to-semantics linking in RRG" (Cortés Rodríguez and Mairal-Usón 2016, 89).

The paper "Building an RRG computational grammar" (Cortés Rodríguez and Mairal-Usón 2016) showcases the creation of rules for NUC of simple clauses in English with some comments about Spanish. This solution aims to contribute to the system ARTEMIS (Automatically Representing Text Meaning via an Interlingua-Based System), initially proposed by Periñán-Pascual (2013) and Periñán-Pascual and Arcas-Túnez (2014).

There is one significant similarity between ARTEMIS and XMG: both frameworks use attribute-value pairs to encode features of nodes[46]. There is a complex system for storing attributes and their values; see (131) (reproduced from Cortés Rodríguez and Mairal-Usón (2016, Fig. 11, p. 99)).

(131) a. Syntactic rules: `X → Y`

b. Attributes as an AVM: `Y [attribute1, attribute 2, ...]`

c. Values of an attribute as an AVM: `A2 [value-a, value-b, ...]`

d. Lexical rules: `token [A1=value-d, A7=value-b, ...]`

Each category participating in a syntactic rule (designed as context-free grammar rules) is assigned a set of attributes (consider (a) and (b) in (131)). As it gets visible from the examples (Cortés Rodríguez and Mairal-Usón 2016, 105), not all of the attributes are obligatory for an instance of a category (a word), so this list tells what can and what cannot be assigned as a feature to a node of a given category. One of the examples given in the paper is the category `MODD` (standing for Modal Auxiliary verb – Deontic). A separate XML structure is created for each attribute, listing all possible values it can take (see (c) in (131)). Finally, each token is assigned a subset of the allowed arguments, each paired with allowed values.

To compare with my XMG-based solution, the structures like (b) in (131) correspond to the definition of typed features in the grammar preamble. In contrast, in my solution, there is no structure that specifies which features can / cannot appear at a given node. On the one hand, this kind of structure in XMG could have become an additional check for errors in the feature structures assigned to nodes. On the other hand, the predefined list of features limits the grammar's flexibility and thus makes it more difficult to introduce specific constraints on individual nodes.

The rewriting rules in ARTEMIS are formulated as rewriting of feature structures. Thus, the initially formulated rule (132a) becomes rewritten as several rules (quite difficult to read, though), one of which is illustrated in (132b).

(132) a. `NUC → MODD PRED` Cortés Rodríguez and Mairal-Usón (2016, 98)

b. `NUC [asp: ?, concept: ?, illoc:?, mod: ?, num:?, per:?,`

[46]For ARTEMIS, the AVMs (attribute-value matrices) are written in XML, and the rest of the solution – in C# (Cortés Rodríguez and Mairal-Usón 2016, 99, 96).

```
recip:?, reflex:?, tpl:?, t: ?] → MODD[illoc: dec, mod:
abl | obl | perm | psbl | vol, num: pl | sg | null, per:
1 | 2 | 3, null t: past | pres | null] AUX [asp: pf |
pr] PRED[concept: ?, recip:?, reflex:?, tpl:?] ‖ MODD
[illoc:dec, mod: abl | obl | perm | psbl | vol, num:
pl | sg | null, per: 1 | 2 | 3, null t: past | pres |
null] AUX [asp: pf] APAR[asp: pr] PRED[concept: ?, recip:?,
reflex:?, tpl:?]
```
Cortés Rodríguez and Mairal-Usón (2016, 109)

After getting used to XMG structures, these rules seem to lack modularity. If each category (namely, MODD) is stored separately together with the feature structure that it must take, why write the whole feature structure in the rule? If the rule requires using a different feature structure than the one stored in the XML entry, how are additional constraints and the context where they apply visible in the final rule?

In terms of approaches towards tree processing, not much detail is given in Cortés Rodríguez and Mairal-Usón (2016). It is stated that parsing is done linearly, word by word (Cortés Rodríguez and Mairal-Usón 2016, 101), but the exact mechanism is not described. When describing syntactic trees, the authors mention that argument structure constructions are stored in the Grammaticon (Cortés Rodríguez and Mairal-Usón 2016, 95). However, there is no information on how these constructions make their way to this module.

In Cortés Rodríguez and Mairal-Usón (2016, 101), the authors follow Periñán-Pascual (2013, 214) and Periñán-Pascual and Arcas-Túnez (2014, 171–175) in postulating additional layers in the RRG structure of the clause. Basic verbal frames – intransitive, transitive, ditransitive – are considered *kernel* constructions, and all other constructions are called *non-kernel* or *L1*. The authors highlight the importance of this difference: "The introduction of a distinction between Kernel and Non-kernel (or L1 Constructions) is of tantamount importance for the design of the parsing rules in the GDE, as it involves the introduction of the CONSTR-L1 node which occupies an intermediate layer between the CORE and the CLAUSE nodes in the LSC" (Cortés Rodríguez and Mairal-Usón 2016, 95).

Perhaps this modification was required because the classical RRG is not quite formalized. If so, the formalization by Osswald and Kallmeyer (2018) is more elegant, as it does not postulate any additional layers, and also more powerful, as it has been proven to apply to a variety of constructions across many different languages.

11.3.3 LinGO Grammar Matrix

A comparison of a grammar engineering solution to existing alternatives could never be complete without the LinGO Grammar Matrix (Bender, Flickinger, and Oepen 2002; Bender, Drellishak, et al. 2010) – perhaps, the best-known and widest-developed grammar engineering project. In this section, I first give a brief account of the overall architecture of the Grammar Matrix and compare it with the overall capacity of an XMG architecture. Next, I give a more detailed review of the library (i.e., a module) of the Grammar Matrix responsible for encoding valency-change morphology.

The essential point of the Grammar Matrix is the possibility of rapid development of custom grammars. It means that a person wishing to develop a grammar of a novel language needs to enter the language's individual properties into a specifically designed questionnaire, which, when combined with the so-called *core grammar*, yields the full, customized grammar of the given language. The core grammar is complemented with the so-called *stored analyses*, contributed by various libraries.

The first difference between XMG projects and the Grammar Matrix is the underlying grammar formalism. The latter is based on HPSG, where, for instance, no syntactic trees in the custom tree-form are used, Hence, there are numerous reasons why tree-based descriptions might be preferable. For example, most syntactic trees are well understandable without much prior knowledge by researchers with different backgrounds (in contrast to HPSG structures, which require some training). However, HPSG has become quite popular and may be preferred by some project leaders.

As for the overall structure, one can regard the Grammar Matrix as an infrastructure yet to be achieved by the XMG community. Currently, there are numerous XMG projects that rely on different grammar formalisms and describe different (mostly individual) languages. It is, however, theoretically possible to join all projects using the same grammar formalism and written in XMG into an umbrella structure that can later serve as a starting point for future projects. This initiative might, however, prove not quite useful, since the existing XMG projects have their individual scopes (e. g., morphology in Ikota, multiple word expressions in French, verbal prefixes in Russian) that are not comparable (Magnana Ekoukou 2014; Savary et al. 2020; Zinova 2017). To put it another way, the Grammar Matrix aims to create comparable grammars that sketch the main traits of languages. In contrast, XMG projects mainly focus on in-depth investigations of individual language phenomena.

This comparative dimension is incorporated into my solution. My solution, which contains well-separated linguistic generalizations and language specifications, can be used as a set of stored analyses of causative constructions and can be easily expanded with new language plugins. The same set of features is required for each language plugin; hence, their values can be elicited from future grammar engineers using a questionnaire, as in the Grammar Matrix.

It is thus logical to compare my solution to the library that stores the same types of analyses in the Grammar Matrix. Such a library has been developed by Christian Curtis (Curtis 2018; Curtis 2019). The library is claimed to account for all valency-changing operations and is evaluated using the material of eight typologically varied languages, some of which overlap with the languages I use in my solution (Lakhota, Japanese, Zulu, Tsez, West Greenlandic, Awa Pit, Javanese). In their analysis, the author of the library relies mainly on Dixon (2000) (which I reconsider in Section 5.1) and Haspelmath and Müller-Bardey (2004). None of these sources is written in HPSG, so the added value of Curtis (2018) lies in its analysis of valency-changing phenomena using HPSG.

The author focuses on the following phenomena: subject-removing and subject-demoting operations, object-removing operations, object-adding operations, subject-adding operations (Curtis 2018, 22). All these operations are considered syntactic, and the author explicitly does not engage with any roles (however, roles are discussed when rules for applicatives and some causatives are formulated). I am not going to analyze the HPSG analyses offered in Curtis (2018). Generally, they are claimed to reflect well the typology offered in Dixon (2000) and Haspelmath and Müller-Bardey (2004). However, even in the evaluation phase, Curtis (2018, 39) noted that the initially proposed analyses do not adequately describe the test sample of languages. Hence, the theoretical part of Curtis (2018) could be improved, at least in this respect.

The implementation suggested in Curtis (2018) is organized as "building blocks" (Curtis 2018, 50), which proves useful for describing multiple languages. These building blocks roughly correspond to elementary trees of XMG. Curtis (2018, 42–50) suggests four blocks operating with the arguments (in HPSG terms, changing the COMPS list). These blocks correspond to the effects of different valency-change operations: subject removal, subject demotion, object removal, and object promotion. There are two special blocks for applicatives: one for adding a new predication and the other for adding arguments in the position specified by the user and co-indexing them. In a more functionalist / constructionist approach, I suppose one could regard the first block as defining the predicate and the second – as defining a construction. An additional block is designed to handle non-local dependencies.

As for causatives, there are two building blocks. The first rule called `scopal-N-place-rel` connects two events and ensures the correct indexing of arguments. This rule can account for two- and three-place causatives and "allows the causative to be modeled as either a raising or control predicate" (Curtis 2018, 48). Even though it is difficult to draw exact parallels between the HPSG-based solution and a tree-based one, this block seems to be parallel to the class `Causative Verbal Stem` as described in Section 10.4.3.2. In my solution, this class accounts for two-, three-, and four-place causatives. The notions of raising and control are irrelevant in RRG. However, in my solution, the class `Causative Verbal Stem` does not regulate the linking of arguments and hence offers the flexibility in this respect (which can be somehow parallel to handling both raising and control at once).

The second rule for causatives suggested in Curtis (2018, 48) is `causative - to - argNofM - tr - op`. Despite the complicated name, the goal of the class is straightforward: it tells "where on the COMPS list the erstwhile subject should be moved" (Curtis 2018, 48). Similar to the pair of rules for applicatives, the first causative rule builds the internal structure of the predicate, and the second rule determines its argument structure. In my solution, the parallel class would be `All Range Causative Constructions`, even though its coverage is much broader.

Curtis (2018, 50–51) illustrates the usage of their solution by "a hypothetical language with a three-place causative expressed by a verbal affix on underlyingly transitive verbs". The lucky coincidence of this example with the scope of my project facilitates the comparison. The entry point of the solution presented in Curtis (2018) is the questionnaire that is presented to the user. The author does not provide the full questionnaire with all possible options; hence, one can only deduce the design from the illustration provided in Curtis (2018, Fig. 5.1, p. 40) and the answers (133) provided for the illustrated situation.

(133) a. Subject-adding operation

b. Should apply to **transitive** targets

c. Erstwhile subject is **at the front** of the complements list

d. The added argument must be **an NP**

e. The added EP should have a predicate name of "**causative**"

The first question determines the type of the modeled construction. Presumably, not all the questions are the same for all the types of verbal derivations. The second question determines which base verbs can participate in the modeled derivation. According to Curtis (2018, Fig. 5.1, p. 40), this question allows selecting one option from a dropdown box. It is not clear whether an option for selecting both intransitive and transitive "targets" exists. The third question determines how the demotion of the causee is realized: different positions within the COMPS list (which, in HPSG, lists the verb's participants) are possible choices. The fourth question determines the category of the added argument. Once again, the list of options is not given, but apparently, a clausal complement requires a different approach. The final question asks for the name the user would like to give the newly defined rule, providing the data for the `predicate - name` block.

The answers to these questions "parametrize" the general rules described in the building blocks. It means they instantiate only the variants relevant to a given language. Once paralleled with my solution, the answers to the questionnaire accommodate the same needs as my `Language Plugins`. Both these structures parametrize – in other words, constrain or shape – the abstractions used to process sentences in a real language.

Curtis (2018, 50) argues that there are five "rule type variants" generated by the answers to the questionnaire given in (133). There are two rules called `scopal - basic - scopal - rel - lex - rule` and `scopal - 3 - place - rel - lex`

`- rule`. The second rule is a specification of the first one. Therefore, one can regard the second rule as crucial for describing the construction in question and the first one as technically auxiliary. The next pair of rules determines the placement of arguments: the demotion of the causee to the second position and the addition of the causer NP as the new subject. The fifth rule is called `causative - tr - lex - rule` and is a conjunction of all the previous rules together with giving the predication the name "causative". The author notes that "[it is] the rule name the user specified in the questionnaire and is the rule type that will be instantiated by the rule instances with their associated morphology" (Curtis 2018, 51). Thus, this fifth rule can be regarded as the final rule that describes the entire causative construction. It is organized as a set of conjunctions and an interface-like name, conceptually similar to the architecture of the `Intersection` classes, despite significant differences in the overall architecture. Apparently, this rule is visible to other libraries, namely, those responsible for morphology. The author does not provide much detail on exactly how the newly developed library, responsible for valency-related operations, shares information with other modules. The idea of keeping "valence change-specific logic localized as much as possible to the valence change library proper" is totally understandable. As a reader external to the Grammar Matrix project, I would have been grateful for more details on the interplay among its different parts. Since my solution is much smaller, I describe the lexical files and their contribution to the main metagrammar in great detail.

The parallelism between my solution and the architecture of the library defined for the Grammar Matrix is remarkable. However, I did not account for this architecture while developing my solution. Therefore, I believe that the similarity of the designs of both solutions is due to the similarity of the goals (which were to describe a given grammar phenomenon in multiple languages).

Chapter 12

Summary

12.1 Summary of computational contributions

In Chapter 10, I presented a solution for parsing correct sentences with three arguments and a predicate with an overt causative affix. This solution is written in the XMG language, and parsing is handled by the TuLiPa parser.

My project is the first XMG project to describe a single grammar phenomenon across multiple languages, offering a clear separation of language-specific information and cross-linguistic generalizations. To store information about individual languages, I designed a special class type, `Language Plugins`, and ensured its import into the classes that describe the structure of an individual sentence. One can trace some similarities between the architecture of Language Plugins and the role of the customization questionnaire in a successful multilingual grammar-engineering project, the LinGO Grammar Matrix. Since the architecture of the Grammar Matrix has not been taken into account during the development of language plugins, this parallelism might reflect that such an architecture is indeed optimal for a truly multilingual project.

One of my suggestions is the idea of `Intersection` classes that combine tree descriptions with language-specific information. Their simple architecture allows them to be created automatically when new languages are added. Moreover, these classes serve as families for anchoring lexical morphemes, ensuring that a sentence in one language is parsed using the lexicon and the plugin for that same language. This additional checkup is important in a multilingual project, especially if it gets updated.

In the syntactic dimension, I also suggest a separation of descriptions that can be reused in other classes from unique descriptions. This architecture reflects the linguistic analysis and is consistent with the general concept of the layered structure of the clause in RRG. It also creates a setup for further expansions of the project.

When describing causative constructions, I have developed a modular yet economical architecture. The separation of each module is motivated by entities of real language data or by steps of linguistic analysis. As a result, the solution I suggest already generates models for describing not only the target sentences but also a range of similar constructions.

Since my solution models only argument structures without adjuncts or operators, I used sister adjunction to add inflectional morphemes. This shortcut can be reused in further projects, also those using sister adjunction of syntactic constituents.

In my solution, I used two lexical dimensions – `morpho` and `lemma`. I suggested a systematic description of various types of morphemes, separating the information required in each class into an abstract part (common to all classes of this kind) and a part specific to each morpheme. Some of the abstractions I created can be reused in further projects. The overall idea of creating morpheme abstractions proves useful, especially in large multilingual projects.

Throughout my work, I reflected on some limitations of the XMG language. Namely, I reproduced morphology as syntax and offered some reflections on whether XMG needs a specific morphosyntactic dimension for morphologically rich languages.

Moreover, I used previously created linguistic examples as input to the parser and raised the question of whether XMG needs its own tool for segmenting words into morphemes.

Using the two logical operations available in XMG – conjunction and disjunction – I handle more complex cases requiring negation and implication. This approach can be reused in further projects. However, if more XMG projects covering similar domains are created, introducing some form of negation or disjunction would make sense.

To overcome the lack of arithmetic addition, I have designed a class describing the operation of valency increase using a disjunction of conjunctions. However, if a larger project requires more similar classes, managing their interaction could become extremely challenging.

Thus, I suggest a novel solution that brings new insights not only in describing causative constructions but also in using XMG and TuLiPa for complex multilingual projects comprising a variety of languages, morphemes, and grammatical phenomena.

12.2 Final remarks and further questions

In this thesis, I have suggested a novel RRG analysis of constructions with a morphological causative derived from a transitive verb and enumerated possible argument-marking strategies. I implemented my theoretical findings by formalizing RRG as a Tree-Wrapping Grammar and provided a solution for parsing real sentences across multiple languages.

In Chapter 2, I have presented RRG and hope to have brought it closer to RRG-external readers. In Chapter 3, I have presented non-RRG studies and hope to have brought them closer to RRG. Even though my thesis contributes to a very specific theoretical framework, I believe that my approach can be reproduced by scholars coming from other theoretical fields or in a theory-independent environment.

In Chapter 5, I have developed a method for studying argument patterns in derived constructions by relating them to other argument patterns attested in the same language. The ways of argument marking are defined in functional terms.

This method proves useful for analyzing and classifying causative constructions in individual languages. Namely, in Chapter 6, I discovered that some languages previously classified as having marginal or uncommon causative constructions can be reclassified as more common types. On the other hand, I demonstrated that previous analyses were obscured by the lack of attention to the functions of the instrumental(-like) case in some languages.

Relying on my theoretical findings, I have formulated language- and construction-related conclusions in Chapter 7. In addition to their theoretical value, they appear useful for developing a solution to parsing causative sentences.

In Part III, namely, in Chapter 10, I presented my solution written in XMG language and compatible with the TuLiPa parser. I proposed a multilingual architecture in which cross-linguistic generalizations are kept separate from language-specific information. In Chapter 11 and Appendix C, I have shown that this solution can parse causative constructions across multiple languages, including those not used in developing the implementation.

The suggested analysis and implementation entail further research steps. From a quantitative typological perspective, one could test the distribution of strategies listed in Chapter 7 on a typologically representative language sample to assess the relative frequency of each strategy. A non-quantitative, in-depth study of causatives in languages with (various types of) syntactic ergativity could yield new insights into both causatives and ergativity. The overall method of relating constructions with an equal number of arguments can be used for studying other valency-increasing constructions (applicatives, version) as well as diatheses in general.

As for implementation, it has the potential to serve as a starting point for a large multilingual XMG-based project (or a set of projects) similar to the LinGO Grammar Matrix. New languages can be easily added to the existing solution. Other grammatical phenomena can be described in the same way and either integrated into the existing project or function as standalone solutions.

Finally, the theoretical and computational results of this thesis contribute to advancing and formalizing RRG.

Appendix A

The code for the main MetaGrammar

```
%% beginregion @Preamble

type CAT = {sentence, clause, core, np, arg, nuc, root,
   aff, verb, nroot, vroot, causaff, casemark, aff, noun,
   stem, core_n, nuc_n, argaff, ecs, naff}
type MR = {actor, ug, nmr}
type CASE = {abl, abs, acc, adess, all, dat, dat-acc, det,
   dir, erg, ins, nom, obl, par}
type TRANS = {intr, tr, ditr}
type MARK = {subst, anchor, star, ddaughter}
type BOOL = {yes, no}
type WO = {sov, svo, vso, vos, ovs, osv}
type VALNC = [0 .. 4]
type VALTYPE = {valincr, valdecr, valmod}
type LOCUS = {head, dep}

feature cat : CAT
feature mr : MR
feature case : CASE
feature wordorder : WO
feature trans : TRANS
feature valnc : VALNC
feature valtype : VALTYPE
feature locusofmarking : LOCUS

feature caus : BOOL
feature _lang_ : LANG
```

```
feature likeditr : BOOL
feature likedemag : BOOL
feature peri : BOOL

property mark : MARK

%=========================%

type LABEL !

feature i : LABEL
feature psa : LABEL
feature causer : LABEL
feature causee : LABEL
feature affectee : LABEL
feature theme : LABEL
feature agent : LABEL
feature recipient : LABEL
feature verb : LABEL
feature e : LABEL
feature eaff : LABEL
feature ecore : LABEL
feature eroot : LABEL

%=========================%
frame-types = {causation, cause, effect, activity, event}
frame-constraints = {cause -> effector:+}

%==========================================================================%

type LANG = {ADY, KBD, BAK, BXK, ESS, EST, FIN, HEB, HUN,
   KAN, KIN, KJH, MTT, NIV, TEL, TPY, TUR, TYV, XAL, YAQ}



feature BoolDoublePsa : BOOL
feature BoolDoubleDo : BOOL
feature BoolMorphDoubleDo : BOOL
feature BoolIncomplDoubleDo : BOOL
```

```
feature BoolCauseeLikeRecip : BOOL
feature BoolCauseeLikeRecip : BOOL
feature BoolCauseeLikePassDemAg : BOOL
feature BoolCauseeLikeCvbAg : BOOL
feature BoolCauseeLikeInstrument : BOOL
feature BoolCauseeLikeUg : BOOL
feature BoolCauseeSpecial : BOOL

feature MorphPsaCase : CASE
feature MorphUgCase : CASE
feature MorphAgCase : CASE
feature MorphRecipCase : CASE
feature MorphPassDemAgCase : CASE
feature MorphCvbAgCase : CASE
feature MorphCauseeCase : CASE
feature MorphInstrCase : CASE
feature MorphDomCase : CASE
feature MorphUnmCase : CASE

feature UnmarkedCases : CASE

feature objectbeforeverb : BOOL
feature subjectinitial : BOOL
%% endregion
%==================================================================%


include LanguagePlugins.mg


%==================================================================%

%%beginregion @Affixes
%==================================================================%

%==============================%
class AnchorOneMorpheme
{<syn>{ node (mark = anchor)[] }}
%==============================%
```

```
class CausAffix
  export ?CAUSAFF ?eaff ?F0 ?Caus ?Valtype
  declare ?CAUSAFF ?eaff ?F0 ?Caus ?Valtype
  { <syn>{ node ?CAUSAFF (mark = subst)[cat = causaff, e =
     ?eaff, caus = ?Caus, valtype = ?Valtype] };
     %%VG:comes from lemma
    <frame>{?eaff[activity,
         effector: ?F0 ]
        }
  }
%===========%
class AnchorAdjunctVerbalInflectAffix
  export ?V ?VerbalInflectAffix
  declare ?V ?VerbalInflectAffix
  {
   <syn>{ node ?V (mark = star) [cat = verb];
      node ?VerbalInflectAffix (mark = anchor)[cat = aff];
      ?V -> ?VerbalInflectAffix}
  }

%===========%
class AnchorAdjunctNominalNoncaseAffix
  export ?N ?NominalNoncaseAffix
  declare ?N ?NominalNoncaseAffix
  {
   <syn>{ node ?N (mark = star) [cat = noun];
      node ?NominalNoncaseAffix (mark = anchor)[cat =
         naff];
      ?N -> ?NominalNoncaseAffix}
  }


%% endregion
%==================================================================%

%% beginregion @Nominals
%==================================================================%
class NounWithCase
```

```
  %%VG: Overgenerates: makes both disjoint varieties for marked and unmarked cases
  import NominalSpine[]
  export ?NodeNoun ?Case ?LanguageFeatures
  declare ?NodeNRoot ?NodeCaseMarker ?Case ?LanguageFeatures
  { <syn> {
    node ?NodeNRoot (mark = anchor) [cat = nroot];
    ?NodeNoun -> ?NodeNRoot;
    {
      { node ?NodeCaseMarker (mark = subst) [cat = casemark, case = ?Case];
      ?NodeNoun -> ?NodeCaseMarker; ?NodeNRoot >> ?NodeCaseMarker}
    |
      { ?Case = ?LanguageFeatures.UnmarkedCases }
    }
  }}

%======================================================================%

class NominalSpine %%aka NP Spine
  %% import NounWithCase[]
  export ?NodeNoun ?NP ?Case ?Mr ?I ?LanguageFeatures
  declare ?NodeNoun ?NP ?CORE_N ?NUC_N ?Case ?Mr ?I ?LanguageFeatures
  {
  <syn> {
       node ?NP [cat = np, case = ?Case, mr = ?Mr, e = ?I] ;  %% TODO: at which level do features mr and i appear?
       node ?CORE_N [cat = core_n, case = ?Case, mr = ?Mr, e = ?I] ;
       node ?NUC_N [cat = nuc_n, case = ?Case, mr = ?Mr, e = ?I] ;
       node ?NodeNoun [cat = noun, case = ?Case] ;
       ?NP -> ?CORE_N ; ?CORE_N -> ?NUC_N ; ?NUC_N -> ?NodeNoun
      }
```

```
  }

%% endregion
%======================================================================%

%% beginregion @Verbs
%======================================================================%
%=========================%
class ValIncrease
export ?Valtype ?Trans ?Valnc
declare ?Valtype ?Trans ?Valnc
{ ?Valtype = valincr;
 {?Trans = intr; ?Valnc = 2} | {?Trans = tr; ?Valnc = 3} |
    {?Trans = ditr; ?Valnc = 4}
}

%=========================%
class VerbalRoot
  export  ?VROOT ?Trans ?F1 ?F2 ?F3 ?eroot
  declare ?VROOT ?Trans ?F1 ?F2 ?F3 ?eroot
  {
  <syn>{node ?VROOT (mark = anchor) [cat = vroot, trans =
     ?Trans, e = ?eroot]};
  <frame>{ {?Trans = intr;
       ?eroot[event,
       participant: ?F1]
       }|
      {?Trans = tr;
       ?eroot[event,
           agent: ?F1,
           patient: ?F2]
       }|
       {?Trans = ditr;
        ?eroot[event,
           agent: ?F1,
           recipient: ?F2,
           theme: ?F3]
       }
    }
```

```
  }
%========================%
class CausativeVerbalStem
  import VerbalRoot[] CausAffix[] ValIncrease[] VerbalSpine[]
  %%VG: We know that causative is a valnc-increasing operation, so we import this class. It double-checks the affix, becaus valtype must unify
  export ?VSTEM ?Trans ?Caus ?everb ?F0 ?F1 ?F2 ?F3 ?Valnc ?Valtype ?V
  declare ?VSTEM ?Trans ?Caus ?everb ?Valnc ?Valtype ?V
  {
  <syn> { node ?VSTEM [cat = stem, e = ?everb, trans = ?Trans, valnc = ?Valnc, valtype = ?Valtype, caus = ?Caus];
      ?VSTEM -> ?VROOT ; ?VSTEM -> ?CAUSAFF; ?V -> ?VSTEM
      %%; ?VROOT >> ?CAUSAFF
    };
  <frame> { ?everb[causation,
          cause : ?eaff,
          effect : ?eroot ]
      }
  }


%========================%
%% class CausativeVerbDep
  %% import CausativeVerbalStem[] VerbalSpine[] %%VerbalInflectAffix[]
  %% export ?V ?Caus
  %% declare ?V
  %% {
  %% <syn> {
      %% node ?V [cat = verb, e = ?everb, valnc = ?Valnc, caus = ?Caus];
      %% ?V -> ?VSTEM; ?NUC -> ?V
    %% }
  %% }
```

```
class VerbalSpine
  export ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V ?everb
  declare ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V ?everb
  {
  <syn> {
       %% node ?SENTENCE [cat = sentence];
       node ?CLAUSE [cat = clause];
       node ?CORE [cat = core, e = ?everb, valnc = ?Valnc, caus = ?Caus];
       node ?NUC [cat = nuc, e = ?everb, valnc = ?Valnc, caus = ?Caus] ;
       node ?V [cat = verb, e = ?everb, valnc = ?Valnc, caus = ?Caus];
       %% ?SENTENCE -> ?CLAUSE;
       ?CLAUSE -> ?CORE; ?CORE -> ?NUC
       ; ?NUC -> ?V
      };
  <iface> {[]}
  }

%% endregion
%====================================================================%

%%beginregion @Headmarking

%====================================================================%
class ArgumentHM
  export ?CLAUSE ?ARGAFF ?ECS ?NP ?ARG ?Case ?eargaff ?Mr
  declare ?CLAUSE ?ARGAFF ?ECS ?NP ?ARG ?Case ?eargaff ?Mr
  { <syn> {
    node ?CLAUSE [cat = clause];
    node ?ARG (mark = ddaughter)[cat = argaff, case = ?Case, e = ?eargaff, mr = ?Mr]; %%two argaff nodes because only one mark per node allowed
    node ?ARGAFF (mark = anchor)[cat = argaff, case = ?Case, e = ?eargaff, mr = ?Mr];
```

```
      ?ARG -> ?ARGAFF;
      {
        {node ?ECS [cat = ecs, e = ?eargaff];
         node ?NP (mark = subst) [cat = np, case = ?Case, e = ?eargaff, mr = ?Mr]; %% feature [e = ?eargaff] to link the two parts of the wrapping tree
         ?CLAUSE -> ?ARG;
         ?CLAUSE -> ?ECS; ?ECS -> ?NP
        }
        |
        {?CLAUSE -> ?ARG}
      }
      }
    }
%================================%
class CausOfTransHM
  import CausativeVerbalStem[] ObjectBeforeVerb[]
  export ?V ?Caus ?Valnc ?ARGAFF_Causer ?ARGAFF_Causee ?ARGAFF_Affectee ?NUC
  declare ?V ?ARGAFF_Causer ?ARGAFF_Causee ?ARGAFF_Affectee ?NUC
  { <syn> { %%node ?V [cat = verb, e = ?everb, valnc = ?Valnc, caus = ?Caus];
        %%?V -> ?VSTEM;
        %%?NUC -> ?V;
      {
        { ?Valnc = 2;
          node ?ARGAFF_Causee (mark=subst) [cat = argaff];
          node ?ARGAFF_Causer (mark=subst) [cat = argaff];
          ?V -> ?ARGAFF_Causee; ?V -> ?ARGAFF_Causer;
          ?ARGAFF_Causer >> ?ARGAFF_Causee; ?ARGAFF_Causee >> ?VSTEM %% THIS IS LANGUAGE-DEPENDENT!
         }
      |
        { ?Valnc = 3;
          node ?ARGAFF_Affectee (mark=subst) [cat = argaff, case = ?LanguageFeatures.MorphUgCase, e = ?F2, mr = ug];
          node ?ARGAFF_Causee (mark=subst) [cat = argaff,
```

```
            case = ?LanguageFeatures.MorphRecipCase, e =
            ?F1, mr = nmr];
         node ?ARGAFF_Causer (mark=subst) [cat = argaff,
            case = ?LanguageFeatures.MorphAgCase, e =
            ?F0, mr = actor];
         ?V -> ?ARGAFF_Causer; ?V -> ?ARGAFF_Causee; ?V
            -> ?ARGAFF_Affectee;
         ?ARGAFF_Causer >> ?ARGAFF_Causee; ?ARGAFF_Causee
            >> ?ARGAFF_Affectee; ?ARGAFF_Affectee >>
            ?VSTEM %% THIS IS LANGUAGE-DEPENDENT!
       }
     }
   }
 }
%===================================================================%
%% class VerbalSpineHM
  %% export ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V
  %% declare ?SENTENCE ?CLAUSE ?CORE ?NUC ?Valnc ?Caus ?V
  %% {
  %% <syn>  {
       %% node ?CLAUSE [cat = clause];
       %% node ?CORE [cat = core, valnc = ?Valnc, caus =
          ?Caus];
       %% node ?NUC [cat = nuc, valnc = ?Valnc, caus =
          ?Caus] ;
       %% ?CLAUSE -> ?CORE; ?CORE -> ?NUC
      %% };
  %% <iface> {[]}
  %% }

%% class ECS_Adjunction
  %% export ?CLAUSE ?ECS ?NP ?NodeNoun ?Case
  %% declare ?CLAUSE ?ECS ?NP ?NodeNoun ?Case
  %% { <syn> { node ?CLAUSE (mark=star) [cat=clause];
        %% node ?ECS [cat=ecs];
        %% node ?NP [cat=np, case = ?Case];
        %% node ?NodeNoun (mark=subst) [cat = noun, case =
           ?Case];
        %% ?CLAUSE -> ?ECS; ?ECS -> ?NP; ?NP -> ?NodeNoun
```

```
      %% }
  %% }

%%endregion
%====================================================================%


%% beginregion @Constructions
%====================================================================%
class AllRangeCausConstruction
import CausativeVerbalStem[]
export ?CORE ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?NP_Recipient ?CauseeCase ?AffecteeCase
declare ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?NP_Recipient ?CauseeCase ?AffecteeCase
{?Caus = yes;
<syn>{node ?NP_Causer (mark = subst) [cat = np, case = ?LanguageFeatures.MorphPsaCase, mr = actor, e = ?F0];
    ?CORE -> ?NP_Causer;
    {
    { ?Valnc = 2; %%Generates 6 correct models
        node ?NP_Causee (mark = subst) [cat = np, case = ?LanguageFeatures.MorphUgCase, mr = ug, e = ?F1];
        ?CORE -> ?NP_Causee
      } |

      { ?Valnc = 3; %%Generates 24 correct models
        node ?NP_Causee (mark = subst) [cat = np, case = ?CauseeCase, e = ?F1, mr = nmr]; %%TODO: check if you have right to determine MRs here
        node ?NP_Affectee (mark = subst) [cat = np, case = ?AffecteeCase, e = ?F2, mr = ug];
        ?CORE -> ?NP_Causee; ?CORE -> ?NP_Affectee
      }
      %% |
      %% { ?Valnc = 4; %%Generates 144 models, two nmr args collapse
        %% node ?NP_Causee (mark = subst) [cat = np, e =
```

```
          ?F1, mr = nmr];
        %% node ?NP_Recipient (mark = subst) [cat = np, e = ?F2, mr = nmr];
        %% node ?NP_Affectee (mark = subst) [cat = np, e = ?F3, mr = ug];
        %% ?CORE -> ?NP_Causee; ?CORE -> ?NP_Recipient; ?CORE -> ?NP_Affectee
      %% }
    }
  }
}

class SimplestCaus
  import AllRangeCausConstruction[]
  export ?CORE ?LanguageFeatures
  {?Trans = intr}

%=========================%
class CausOfTrans
  import AllRangeCausConstruction[] ObjectBeforeVerb[] SubjectInitial[]
  export ?CORE ?LanguageFeatures
  declare ?LanguageFeatures ?ToRemove
  { ?Trans = tr;
    ?ToRemove = +;
  <syn>{ ?AffecteeCase = ?LanguageFeatures.MorphUgCase;
             {
      {?LanguageFeatures.BoolCauseeLikeRecip = yes;
        ?CauseeCase = ?LanguageFeatures.MorphRecipCase
      } |
      {?LanguageFeatures.BoolCauseeLikePassDemAg = yes;
        ?CauseeCase = ?LanguageFeatures.MorphPassDemAgCase
      } |
      {?LanguageFeatures.BoolCauseeLikeCvbAg = yes;
        ?CauseeCase = ?LanguageFeatures.MorphCvbAgCase
      } |
      {?LanguageFeatures.BoolCauseeLikeInstrument = yes; %%IMPORTANT: a 2-arg core + peri?
        ?CauseeCase = ?LanguageFeatures.MorphInstrCase
```

```
        } |
        {?LanguageFeatures.BoolCauseeLikeUg = yes; %%TODO: how to add restrictions on marked/unmarked?
          ?CauseeCase = ?LanguageFeatures.MorphUgCase
        }
          }
      }
}
%%endregion
%==================================================================%

%%beginregion @Wordorder
class ObjectBeforeVerb
export  ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?CauseeCase ?AffecteeCase ?MrCausee ?MrAffectee ?F2 ?NUC
declare ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?CauseeCase ?AffecteeCase ?MrCausee ?MrAffectee ?F2 ?NUC
{
<syn>{  {?LanguageFeatures.objectbeforeverb = yes;
    node ?NP_Affectee [case = ?LanguageFeatures.MorphUgCase]; node ?NUC;
    ?NP_Affectee >> ?NUC}
    | {?LanguageFeatures.objectbeforeverb = no}
  }
}

class SubjectInitial
export  ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?CauseeCase ?AffecteeCase ?MrCausee ?MrAffectee ?F2 ?NUC
declare ?LanguageFeatures ?NP_Causer ?NP_Causee ?NP_Affectee ?CauseeCase ?AffecteeCase ?MrCausee ?MrAffectee ?F2 ?NUC
{
<syn>{  {?LanguageFeatures.subjectinitial = yes;
    node ?NP_Causer [case = ?LanguageFeatures.MorphPsaCaseCase];
```

```
    node ?NUC;
    ?NP_Causer >> ?NUC}
    | {?LanguageFeatures.subjectinitial = no}
  }
}
%%endregion
%==================================================================%


%% beginregion @Intersections
%==================================================================%


include intersections.mg

%% endregion
%==================================================================%

%% beginregion  @Valuation

%======= Required for correct parsing ====%
value AnchorOneMorpheme
value AnchorAdjunctVerbalInflectAffix
value AnchorAdjunctNominalNoncaseAffix
value NounWithCase

%======= Enter lang name ====%

include valuations.mg

%% endregion
%==================================================================%
```

Appendix B

Sample lexical metagrammars (example language: Finnish)

B.1 Morpho

```
%====== Abstractions ======%
class Abstr_NRoot{
<morpho>{
  cat <- nroot
}}

class Abstr_VRoot{
<morpho>{
  trans <- tr;
  cat <- vroot
}}

class Abstr_CaseMarker{
<morpho>{
  cat <- casemark
}}

class Abstr_VInflect{
<morpho>{
  cat <- aff
}}

class Abstr_NInflect{
```

```
<morpho>{
  cat <- naff
}}

class Abstr_ArgAffix{
<morpho>{
  cat <- argaff
}}

class Abstr_CausAffix{
<morpho>{
  lemma <- "CAUS";
  caus <- yes;
  valtype <- valincr;
  cat <- causaff
}}
%============%
class NRoot_naapuri
import Abstr_NRoot[]
{<morpho>{
morph <- "naapuri";
lemma <- "neighbour"
 }}

class VInflect_a
import Abstr_VInflect[]
{<morpho>{
morph <- "a";
lemma <- "3Sg"
 }}

class VRoot_kuiva
import Abstr_VRoot[]
{<morpho>{
morph <- "kuiva";
lemma <- "dry"
 }}

class CausAffix_tta
```

```
import Abstr_CausAffix[]
{<morpho> {morph <- "tta" }}

class CaseMarker_lla
import Abstr_CaseMarker[]
{<morpho>{
morph <- "lla";
case <- adess;
lemma <- "Adess"
 }}

class NInflect_nsa
import Abstr_NInflect[]
{<morpho>{
morph <- "nsa";
lemma <- "Poss3"
 }}

class NInflect_an
import Abstr_NInflect[]
{<morpho>{
morph <- "an";
lemma <- "Poss3"
 }}

class NRoot_mikko
import Abstr_NRoot[]
{<morpho>{
morph <- "mikko";
lemma <- "Mikko"
 }}

class NRoot_pyykki
import Abstr_NRoot[]
{<morpho>{
morph <- "pyykki";
lemma <- "laundry"
 }}
```

```
class NRoot_talo
import Abstr_NRoot[]
{<morpho>{
morph <- "talo";
lemma <- "house"
 }}

class CausAffix_utta
import Abstr_CausAffix[]
{<morpho> {morph <- "utta" }}

class CaseMarker_n
import Abstr_CaseMarker[]
{<morpho>{
morph <- "n";
case <- acc;
lemma <- "Acc"
 }}

class VRoot_rakenn
import Abstr_VRoot[]
{<morpho>{
morph <- "rakenn";
lemma <- "build"
 }}

class NRoot_kalle
import Abstr_NRoot[]
{<morpho>{
morph <- "kalle";
lemma <- "Kalle"
 }}

class NRoot_peka
import Abstr_NRoot[]
{<morpho>{
morph <- "peka";
lemma <- "Pekka"
 }}
```

```
class NRoot_diane
import Abstr_NRoot[]
{<morpho>{
morph <- "diane";
lemma <- "Diane"
 }}

class NRoot_satu
import Abstr_NRoot[]
{<morpho>{
morph <- "satu";
lemma <- "Satu"
 }}

class CausAffix_tt
import Abstr_CausAffix[]
{<morpho> {morph <- "tt" }}

class VInflect_i
import Abstr_VInflect[]
{<morpho>{
morph <- "i";
lemma <- "Pst3Sg"
 }}

class NRoot_etana
import Abstr_NRoot[]
{<morpho>{
morph <- "etana";
lemma <- "slug"
 }}

class VRoot_tapa
import Abstr_VRoot[]
{<morpho>{
morph <- "tapa";
lemma <- "kill"
 }}
```

```
value CaseMarker_lla
value CaseMarker_n
value CausAffix_tt
value CausAffix_tta
value CausAffix_utta
value NInflect_an
value NRoot_diane
value NRoot_kalle
value NRoot_mikko
value NInflect_nsa
value NRoot_peka
value NRoot_satu
value NRoot_etana
value NRoot_naapuri
value NRoot_pyykki
value NRoot_talo
value VInflect_a
value VInflect_i
value VRoot_kuiva
value VRoot_rakenn
value VRoot_tapa
```

B.2 Lemma

```
%====== Abstractions ======%
class Abstr_Lemma_NRoot{
<lemma>{
  cat <- nroot;
  fam <- ArgumentFIN
}}

class Abstr_Lemma_VRoot{
<lemma>{
  cat <- vroot;
  fam <- IntersectionFIN
}}
```

```
class Abstr_Lemma_CaseMarker{
<lemma>{
  cat <- casemark;
  fam <- AnchorOneMorpheme
}}

class Abstr_Lemma_VInflect{
<lemma>{
  cat <- aff;
  fam <- AnchorAdjunctVerbalInflectAffix
}}

class Abstr_Lemma_NInflect{
<lemma>{
  cat <- naff;
  fam <- AnchorAdjunctNominalNoncaseAffix
}}

class Abstr_Lemma_ArgAffix{
<lemma>{
  cat <- argaff;
  fam <- AnchorArgumentAffix
}}

class Lemma_CausAffix{
<lemma>{
  entry <- "CAUS";
  cat <- causaff;
  fam <- AnchorOneMorpheme
}}
%============%
class Lemma_NRoot_Mikko
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "Mikko" }}

class Lemma_VInflect_3Sg
import Abstr_Lemma_VInflect[]
{<lemma> { entry <- "3Sg" }}
```

```
class Lemma_VRoot_dry
import Abstr_Lemma_VRoot[]
{<lemma> { entry <- "dry" }}

class Lemma_NRoot_laundry
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "laundry" }}

class Lemma_NRoot_neighbour
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "neighbour" }}

class Lemma_NInflect_Poss3
import Abstr_Lemma_NInflect[]
{<lemma> { entry <- "Poss3" }}

class Lemma_CaseMarker_Adess
import Abstr_Lemma_CaseMarker[]
{<lemma> { entry <- "Adess" }}

class Lemma_NRoot_Kalle
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "Kalle" }}

class Lemma_NRoot_house
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "house" }}

class Lemma_NRoot_Pekka
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "Pekka" }}

class Lemma_CaseMarker_Acc
import Abstr_Lemma_CaseMarker[]
{<lemma> { entry <- "Acc" }}

class Lemma_VRoot_build
import Abstr_Lemma_VRoot[]
{<lemma> { entry <- "build" }}
```

```
class Lemma_NRoot_Satu
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "Satu" }}

class Lemma_VInflect_Pst3Sg
import Abstr_Lemma_VInflect[]
{<lemma> { entry <- "Pst3Sg" }}

class Lemma_VRoot_kill
import Abstr_Lemma_VRoot[]
{<lemma> { entry <- "kill" }}

class Lemma_NRoot_slug
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "slug" }}

class Lemma_NRoot_Diane
import Abstr_Lemma_NRoot[]
{<lemma> { entry <- "Diane" }}

value Lemma_CaseMarker_Acc
value Lemma_CaseMarker_Adess
value Lemma_CausAffix
value Lemma_NRoot_Diane
value Lemma_NRoot_Kalle
value Lemma_NRoot_Mikko
value Lemma_NRoot_Pekka
value Lemma_NInflect_Poss3
value Lemma_NRoot_Satu
value Lemma_NRoot_house
value Lemma_NRoot_laundry
value Lemma_NRoot_neighbour
value Lemma_NRoot_slug
value Lemma_VInflect_3Sg
value Lemma_VInflect_Pst3Sg
value Lemma_VRoot_build
value Lemma_VRoot_dry
value Lemma_VRoot_kill
```

Appendix C

Testing results

The table below illustrates the results of the testing. In the first column, the example sentence is referenced. Sentences that were not cited in the main part of the dissertation are provided below the table. The expected output for each sentence is **1 good tree**. This counts as correct ✔. Sometimes, multiple sentence representations are provided by the parser. Usually, it occurs for sentences where multiple nouns bear unmarked cases. Among these multiple trees, one is correct. These are cases of over-generation ✚. One example in the table has the "??" judgement, which means that it is almost unacceptable for the speakers. Therefore, it should **not** be parse as a correct sentence. However, there is a representation provided. This counts as a false positive ◐.

Example	Language	Result	Output file	Status
(95)	Adyghe	1 good tree	ADY1	✔
(134)	Bashkir	1 good tree	BAK1	✔
(38b)	Bashkir	1 good tree	BAK2	✔
(45)	Bashkir	1 good tree	BAK3	✔
(43)	Bashkir	1 good tree	BAK4	✔
(70)	Lubukusu	1 good tree	BXK1	✔
(72)	Lubukusu	1 good tree	BXK2	✔
(135)	Asiatic Eskimo	1 good tree	ESS1	✔
(33)	Estonian	2 trees	EST1	✚
(54)	Finnish	4 trees	FIN1	✚
(57)	Finnish	1 good tree	FIN2	✔
(55a)	Finnish	1 good tree	FIN3	✔
(138)	Kalmyk	2 trees	XAL1	✚
(46a)	Kalmyk	2 trees	XAL2	✚
(136a)	Kannada	1 good tree	KAN1	✔

(136b)	Kannada	1 good tree	KAN2	✓
(36a)	Kannada	1 good tree	KAN3	✓
(36b)	Kannada	1 good tree	KAN4	✓
(111)	Khakas	1 good tree	KJH1	✓
??(112)	Khakas	1 tree	KJH2	◑
(41)	Mishar Tatar	1 good tree	MTT1	✓
(63)	Nivkh	1 good tree	NIV1	✓
(64)	Nivkh	1 good tree	NIV1	✓
(21)	Telugu	2 trees	TEL1	✚
(137)	Turkish	1 good tree	TUR1	✓
(113)	Tuvinian	1 good tree	TYV1	✓
(114)	Tuvinian	1 good tree	TYV2	✓
(139)	Yaqui	1 good tree	YAQ1	✓

Additional examples

(134) min bala-nan xat jað-ðər-a-m — Bashkir
1sg child-abl letter write-caus-ipfv-1sg
'I make the child write a letter.'□ — Dmitriev 2008 [1948], p. 225

(135) — Asiatic Eskimo
aRna-m qamtə-sta-qaa xLuk panigmi-nun
woman-erg extinguish-caus-3sg:3sg oven(abs) daughter.her-dat

(136) a. Avanu-∅ nana-ge bisket-annu tinn-is-id-anu — Kannada
3sg-nom 1sg-dat biscuit-acc eat-caus-pst-3sg.m
'He fed me a biscuit,' or 'He made me eat a biscuit.'

b. Avanu-∅ nann-inda bisket-annu tinn-is-id-anu
3sg-nom 1sg-ins biscuit-acc eat-caus-pst-3sg.m
'He had me eat a biscuit.' — Van Valin 2018a, p. 127, (17)

(137) Dišči mektub-u müdür-e imzala-t-tı — Turkish
dentist letter-acc boss-dat sign-caus-pst
'The dentist made the boss sign the letter.' — Comrie 1976

(138) ezə-n ködəlməšč-är xö al-ulə-v KALMYK
master-EXT servant-INS ram kill-CAUS-PST
'The master made the servant kill the ram.'□ Say 2009, p. 387, (1)

(139) Aurelia-∅ Karmen-ta usi-ta bit-tua-k YAQUI
Aurelia-NOM Carmen-ACC child-ACC see-CAUS-PSTPFV
'Aurelia showed Carmen the child'. Guerrero and Van Valin 2004, p. 36, (32b)
VG: a determiner and an adjective were removed from the original example (128)

Appendix D

Script for creating XMG lexical files from glossed examples

```
import os
import re

pattern_lang = re.compile(r'(?<=label{ex:)[A-Z]{3}')
pattern_gla = re.compile(r'(?<=textsc).*{[a-zA-Z]*}}')
pattern_chrnum = re.compile(r'(?<=Uchar\").{4}')
pattern_hacek = re.compile(r'(?<=\\v{)[a-zA-Z](?=})')
pattern_macron = re.compile(r'(?<=\\={)[a-zA-Z](?=})')
pattern_acute = re.compile(r"(?<=\\\'{)[a-zA-Z](?=})")
pattern_trema = re.compile(r'(?<=\\\")[a-zA-Z]')

lemma_abstractions_1 = '''%====== Abstractions ======%
class Abstr_Lemma_NRoot{
<lemma>{
  cat <- nroot;
  fam <- Argument'''
lemma_abstractions_2 = '''class Abstr_Lemma_VRoot{
<lemma>{
  cat <- vroot;
  fam <- Intersection'''
lemma_abstractions_3 = '''class Abstr_Lemma_CaseMarker{
<lemma>{
  cat <- casemark;
```

```
  fam <- AnchorCaseAffix
}}

class Abstr_Lemma_VInflect{
<lemma>{
  cat <- aff;
  fam <- AnchorAdjunctVerbalInflectAffix
}}

class Abstr_Lemma_NInflect{
<lemma>{
  cat <- naff;
  fam <- AnchorAdjunctNominalNoncaseAffix
}}

class Abstr_Lemma_ArgAffix{
<lemma>{
  cat <- argaff;
  fam <- AnchorArgumentAffix
}}

class Lemma_CausAffix{
<lemma>{
  entry <- "CAUS";
  cat <- causaff;
  fam <- AnchorCausAffix
}}
%============%

'''
morpho_abstractions = '''%====== Abstractions ======%
class Abstr_NRoot{
<morpho>{
  cat <- nroot
}}

class Abstr_VRoot{
<morpho>{
  trans <- tr;
```

```
  cat <- vroot
}}

class Abstr_CaseMarker{
<morpho>{
  cat <- casemark
}}

class Abstr_VInflect{
<morpho>{
  cat <- aff
}}

class Abstr_NInflect{
<morpho>{
  cat <- naff
}}

class Abstr_ArgAffix{
<morpho>{
  cat <- argaff
}}

class Abstr_CausAffix{
<morpho>{
  lemma <- "CAUS";
  caus <- yes;
  valtype <- valincr;
  cat <- causaff
}}
%============%

'''

dict_numeral_glosses = {
    'First{}': '1',
    'Second{}': '2',
    'Third{}': '3',
    'Cone{}': 'c1',
```

```
    'Ctwo{}': 'c2',
    'Cthree{}': 'c3',
    'Cfour{}': 'c4',
    'Cfive{}': 'c5',
    'Csix{}': 'c6',
    'Cseven{}': 'c7',
    'Ceight{}': 'c8',
    'Cnine{}': 'c9',
    'Cten{}': 'c10',
}
set_case_glosses = {"Abl", "Abs", "Acc", "Add", "Adess",
   "All", "Com", "Dat", "Erg", "Gen", "Ins", "Loc", "Nom",
   "Obj", "Obl", "Par", "Ug", "Unm", "Voc"}


def fun_process_gla(example_line):
    try:
        text =
           example_line[pattern_gla.search(example_line).end():-3]
    except:
        text = example_line[re.match(r"\\gla",
           example_line).end() + 1:-3]

    text = text.replace('€™â', "'")
    text = re.sub(pattern_hacek, r'\g<0>\\Uchar"030C',
       text)
    text = re.sub(pattern_macron, r'\g<0>\\Uchar"0304',
       text)
    text = re.sub(pattern_acute, r'\g<0>\\Uchar"0301',
       text)
    text = re.sub(pattern_trema, r'\g<0>\\Uchar"0308',
       text)
    uchars = set(re.findall(pattern_chrnum, text))
    for uchar in uchars:
        text = text.replace('\\Uchar"{}'.format(uchar),
           chr(int(uchar, 16)))
    text = text.replace('\\v', '')
    text = text.replace("\\\'", '')
    text = text.replace('\\\"', '')
```

```
    text = text.replace("\\\=", '')
    text = text.replace('\\textsuperscript{', '')
    text = text.replace('}', '')
    text = text.replace('{', '')
    words = text.split()
    morphemes = []
    for word in words:
        morphemes.append(tuple(word.split('-')))
    return morphemes


def fun_process_glb(example_line):
    gloss_string = example_line[5:-3]
    gloss_string = gloss_string.replace("Adagl", '')
    for k in dict_numeral_glosses.keys():
        gloss_string = gloss_string.replace(k, dict_numeral_glosses[k])
    gloss_string = gloss_string.replace('{}', '')
    gloss_string = gloss_string.replace('\\', '')
    gloss_string = gloss_string.replace('.', '')
    words_gloss_string = gloss_string.split()
    glosses = []
    for word in words_gloss_string:
        glosses.append(tuple(word.split('-')))
    return glosses


def fun_create_lemma_file():
    lemma_file = open('output/Auto_Lemma_{}.mg'.format(lang), 'w', encoding='utf8')
    lemma_file.write(lemma_abstractions_1 + lang + "\n}}\n\n")
    lemma_file.write(lemma_abstractions_2 + lang + "\n}}\n\n")
    lemma_file.write(lemma_abstractions_3)
    lemma_file.close()
    lemma_lines = open('output/Auto_Lemma_{}.mg'.format(lang), 'r',
```

```
        encoding='utf8').readlines()
    print("File output/Auto_Lemma_{}.mg created".format(lang))
    return lemma_lines


def fun_create_morpho_file():
    morpho_file = open('output/Auto_Morpho_{}.mg'.format(lang), 'w', encoding='utf8')
    morpho_file.write(morpho_abstractions)
    print("File output/Auto_Morpho_{}.mg created".format(lang))


def fun_create_entries(morpheme, gloss, cat):
    lemma_entryline_title = "class Lemma_{}_{}".format(cat, gloss)
    lemma_entryline_import = "import Abstr_Lemma_{}[]".format(cat)
    lemma_entryline_content = '{<lemma> { entry <- ' + '"{}"'.format(gloss) + ' }}\n\n'
    lemma_entry = "\n".join([lemma_entryline_title, lemma_entryline_import, lemma_entryline_content])

    morpho_entryline_title = "class {}_{}".format(cat, ''.join(ch for ch in morpheme if ch.isalnum()).lower())
    morpho_entryline_import = "import Abstr_{}[]".format(cat)
    morpho_entryline_dim = "{<morpho>{"
    morpho_entryline_morph = 'morph <- "{}";'.format(morpheme.lower())

    if cat == "CaseMarker":
        morpho_entryline_case = '\ncase <- {};'.format(gloss.lower())
    else:
        morpho_entryline_case = str("")
```

```
    morpho_entryline_morph += morpho_entryline_case

    morpho_entryline_lemma = 'lemma <- "{}"'.format(gloss)
    morpho_entryline_end = " }}\n\n"
    morpho_entry = "\n".join(
        [morpho_entryline_title, morpho_entryline_import,
          morpho_entryline_dim, morpho_entryline_morph,
          morpho_entryline_lemma, morpho_entryline_end])

    lemma_value = ("value Lemma_{}_{}\n".format(cat,
      gloss))
    morpho_value = ("value {}_{}\n".format(cat, ''.join(ch
      for ch in morpheme if ch.isalnum()).lower()))

    new_lemma_classes.append(lemma_entry)
    new_morpho_classes.append(morpho_entry)
    new_lemma_values.append(lemma_value)
    new_morpho_values.append(morpho_value)


def fun_create_caus_entries(morpheme):
    new_lemma_values.append("value Lemma_CausAffix\n")
    new_morpho_values.append("value
      CausAffix_{}\n".format(''.join(ch for ch in
      morpheme if ch.isalnum())))

    morpho_entry = "class
      CausAffix_{}\n".format(''.join(ch for ch in
      morpheme if ch.isalnum())) + "import
      Abstr_CausAffix[]\n" + '''{<morpho> {morph <- ''' +
      '"{}"'.format(
        morpheme) + ' }}\n\n'
    new_morpho_classes.append(morpho_entry)


directory = 'source'
for filename in os.listdir(directory):
    example = open('{}/{}'.format(directory, filename),
```

```
        'r')
    print("\nProcessing {}\n".format(filename))
    # example = open('source/ex_KBD_Caus_Read.tex', 'r')

    for example_line in example.readlines():
        if re.match(r"\\ex|\\pex", example_line) != None:
            lang =
               pattern_lang.search(example_line).group()
            if
               os.path.exists('output/Auto_Lemma_{}.mg'.format(lang))
               == True:
                # lemma_file =
                   open('output/Auto_Lemma_{}.mg'.format(lang),
                   'a', encoding="utf8")
                with
                   open('output/Auto_Lemma_{}.mg'.format(lang),
                   'r', encoding="utf8") as lemma_reading:
                    lemma_lines = lemma_reading.readlines()
                    l = 0
                    try:
                        while re.match("value",
                           lemma_lines[l]) == None:
                            l += 1
                        lemma_classes = lemma_lines[:l]
                        lemma_values = lemma_lines[l:]
                    except:
                        lemma_classes = lemma_lines
                        lemma_values = []
            else:
                lemma_classes = fun_create_lemma_file()
                lemma_values = []
            if
               os.path.exists('output/Auto_Morpho_{}.mg'.format(lang))
               == True:
                # morpho_file =
                   open('output/Auto_Morpho_{}.mg'.format(lang),
                   'a', encoding="utf8")
                with
                   open('output/Auto_Morpho_{}.mg'.format(lang),
```

```
                 'r', encoding="utf8") as morpho_reading:
                    morpho_lines =
                       morpho_reading.readlines()
                    m = 0
                    try:
                        while re.match("value",
                           morpho_lines[m]) == None:
                            m += 1
                        morpho_classes = morpho_lines[:m]
                        morpho_values = morpho_lines[m:]
                    except:
                        morpho_classes = morpho_lines
                        morpho_values = []
            else:
                fun_create_morpho_file()
                morpho_classes = morpho_abstractions
                morpho_values = []
        elif re.match(r"\\gla", example_line) != None:
            morphemes = fun_process_gla(example_line)
        elif re.match(r"\\glb", example_line) != None:
            glosses = fun_process_glb(example_line)

    print(morphemes)
    print(glosses)

    with open("Sentences_with_Translations.txt", 'a',
       encoding="utf8") as sentences_file:
        sentences_file.write(' '.join([str(x).lower() for
           t in morphemes for x in t]))
        sentences_file.write('\n')
        sentences_file.write('%% ' + str(glosses))
        sentences_file.write('\n\n')

    with open("Sentences_for_Parsing.txt", 'a',
       encoding="utf8") as sentences_file:
        sentences_file.write(' '.join([str(x).lower() for
           t in morphemes for x in t]))
        sentences_file.write('\n')
```

```
    new_lemma_classes = []
    new_lemma_values = []
    new_morpho_classes = []
    new_morpho_values = []

    for i in range(0, len(morphemes)):
        word = morphemes[i]
        word_glosses = glosses[i]
        if 'Caus' in word_glosses:
            # it is a verb
            for j in range(0, len(word_glosses)):
                gloss = word_glosses[j]
                morpheme = word[j]
                if gloss.islower():
                    if "value {}_{}\n".format("VRoot", morpheme) not in morpho_values:
                        fun_create_entries(morpheme, gloss, "VRoot")
                elif gloss == 'Caus':
                    if "value CausAffix_{}\n".format(''.join(ch for ch in morpheme if ch.isalnum())) not in morpho_values:
                        fun_create_caus_entries(morpheme)
                # elif gloss[0] in ('1', '2', '3'):
                #     if "value {}_{}\n".format("ArgAffix", morpheme) not in morpho_values:
                #         fun_create_entries(morpheme, gloss, "ArgAffix")
                else:
                    if "value {}_{}\n".format("VInflect", morpheme) not in morpho_values:
                        fun_create_entries(morpheme, gloss, "VInflect")
        else:
            for j in range(0, len(word_glosses)):
                gloss = word_glosses[j]
                morpheme = word[j]
```

```
                if gloss.islower():
                    if "value {}_{}\n".format("NRoot", morpheme) not in morpho_values:
                        fun_create_entries(morpheme, gloss, "NRoot")
                elif gloss in set_case_glosses:
                    if "value {}_{}\n".format("CaseMarker", morpheme) not in morpho_values:
                        fun_create_entries(morpheme, gloss, "CaseMarker")
                else:
                    if "value {}_{}\n".format("NInflect", morpheme) not in morpho_values:
                        fun_create_entries(morpheme, gloss, "NInflect")

    with open('output/Auto_Lemma_{}.mg'.format(lang), 'w', encoding="utf8") as lemma_writing:
        lemma_writing.write("".join(lemma_classes))
        lemma_writing.write("".join(set(new_lemma_classes)))
        # lemma_writing.write("\n\n")
        lemma_writing.write("".join(sorted(set(lemma_values + new_lemma_values))))

    with open('output/Auto_Morpho_{}.mg'.format(lang), 'w', encoding="utf8") as morpho_writing:
        morpho_writing.write("".join(morpho_classes))
        morpho_writing.write("".join(set(new_morpho_classes)))
        # morpho_writing.write("\n\n")
        morpho_writing.write("".join(sorted(set(morpho_values + new_morpho_values))))
```

References

Ackerman, Farrell, Robert Malouf, and John Moore (2017). "Symmetrical objects in Moro: Challenges and solutions." In: *Journal of Linguistics* 53.1, 3–50.

Ackerman, Farrell and John Moore (1994). "Valence and the Semantics of Causativization." In: *Proceedings of the 20th Annual Meeting of the Berkeley Linguistics Society*. Ed. by Susanne Gahl, Andy Dolbey, and Christopher Johnson. Vol. 20. 1, 1–13.

Ackerman, Farrell and John Moore (1999). "Syntagmatic and paradigmatic dimensions of causee encodings." In: *Linguistics and Philosophy* 22.1, 1–44.

Alsina, Alex (1992). "On the argument structure of causatives." In: *Linguistic inquiry* 23.4, 517–555.

Alsina, Alex and Sam Mchombo (1990). "The syntax of applicatives in Chichewâ: Problems for a theta theoretic asymmetry." In: *Natural Language & Linguistic Theory* 8.4, 493–506.

Anderson, John M. (2005). "The argument structure of morphological causatives." In: *Poznan Studies in Contemporary Linguistics* 40, 27–89.

Andreou, Marios and Simon Petitjean (2017). "Describing derivational polysemy with XMG." In: *Actes des 24ème conférence sur le traitement automatique des langues naturelles. Volume 2 – Articles courts*. Ed. by Iris Eshkol-Taravella and Jean-Yves Antoine, 94–101.

Arkadiev, Peter M., Yury A. Lander, Alexander B. Letuchiy, and Nina R. Sumbatova (2009). "Vvedenie. Osnovnye svedenija ob adygejskom jazyke [Introduction. Basic information on the Adyghe language]." In: *Aspekty polisintetizma: ocherki po grammatike adygejskogo jazyka [Aspects of polysynthesis: Essays in Adyghe grammar]*. Ed. by Yakov G. Testelets. Moscow: RSUH, 17–120.

Arkadiev, Peter M. and Alexander B. Letuchiy (2009). *The syntax and semantics of event structure and Adyghe causatives*. Manuscript posted as Lingbuzz/000811.

Arps, David, Tatiana Bladier, and Laura Kallmeyer (2019). "Chart-based RRG parsing for automatically extracted and hand-crafted RRG grammars." In: *Role and Reference Grammar RRG Conference*. University at Buffalo.

Arps, David and Simon Petitjean (2018). "A parser for LTAG and frame semantics." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga. Miyazaki.

Baker, Mark (1988). *Incorporation: A Theory of Grammatical Function Changing*. Chicago: University of Chicago Press.

Baker, Mark, Ken Safir, and Justine Sikuku (2012). "Sources of (A) Symmetry in Bantu double object constructions." In: *Proceedings of the 30th West Coast Conference on Formal Linguistics*. Ed. by Nathan Arnett and Ryan Bennett. Somerville, MA: Cascadilla Proceedings Project, 54–64.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Laurie Poulson, and Safiyyah Saleem (2010). "Grammar customization." In: *Research on Language and Computation* 8.1, 23–72.

Bender, Emily M., Dan Flickinger, and Stephan Oepen (2002). "The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars." In: *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Ed. by J. Carroll, N. Oostdijk, and R. Sutcliffe. Taipei, Taiwan, 8–14.

Bentley, Delia, Ricardo Mairal Usón, Wataru Nakamura, and Robert D. Van Valin Jr., eds. (2023). *The Cambridge Handbook of Role and Reference Grammar*. Cambridge: Cambridge University Press.

Bickel, Balthasar (2010). "Capturing particulars and universals in clause linkage." In: *Clause Linking and Clause Hierarchy: Syntax and Pragmatics*. Ed. by Isabelle Bril. Amsterdam; Philadelphia: John Benjamins, 51–101.

Bladier, Tatiana, Kilian Evang, Valeria Generalova, Zahra Ghane, Laura Kallmeyer, Robin Möllemann, Natalia Moors, Rainer Osswald, and Simon Petitjean (2022). "RRGparbank: A Parallel Role and Reference Grammar Treebank." In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*. Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, 4833–4841.

Bonch-Osmolovskaja, Anastasija (2007). "Semantika aktantnyx derivacij [Semantics of valence modifications]." In: *Mišarskij dialekt tatarskogo jazyka: Očerki po sintaksisu i semantike [Mishar dialect of the Tatar language: Notes on syntax and semantics]*. Ed. by E. A. Lutikova, K. I. Kazenin, V. D. Solovjev, and S. G. Tatevosov. Kazan: Magarif, 143–191.

Bresnan, Joan and Sam A. Mchombo (1987). "Topic, pronoun, and agreement in Chicheŵa." In: *Language* 63, 741–782.

Bresnan, Joan and Lioba Moshi (1990). "Object asymmetries in comparative Bantu syntax." In: *Linguistic inquiry* 21.2, 147–185.

Candito, Marie (1996). “A principle-based hierarchical representation of LTAGs.” In: *COLING 1996: The 16th International Conference on Computational Linguistics. Proceedings. Volume 1.* Copenhagen.

Candito, Marie (1999). “Représentation modulaire et paramétrable de grammaires électroniques lexicalisées: application au français et à l'italien.” PhD thesis. Université Paris 7.

Clément, Lionel and Alexandra Kinyon (2003). “Generating parallel multilingual LFG-TAG grammars from a MetaGrammar.” In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics.* Sapporo: Association for Computational Linguistics, 184–191.

Colarusso, John (1992). *A grammar of the Kabardian language.* Calgary: University of Calgary Press.

Cole, Peter (1976). “A causative construction in Modern Hebrew: theoretical implications.” In: *Studies in Modern Hebrew syntax and semantics.* Ed. by Peter Cole. Amsterdam: North-Holland Publishing Company, 99–128.

Cole, Peter (1983). “The grammatical role of the causee in universal grammar.” In: *International Journal of American Linguistics* 49.2, 115–133.

Comrie, Bernard (1974). “Causatives and universal grammar.” In: *Transactions of the Philological Society* 73.1, 1–32.

Comrie, Bernard (1976). “The syntax of causative constructions: cross-language similarities and divergences.” In: *Syntax and semantics: The grammar of causative constructions.* Ed. by Masayoshi Shibatani. New York: Academic Press, 261–312.

Comrie, Bernard (2003). “Recipient person suppletion in the verb ‘give’.” In: *Language and life: Essays in memory of Kenneth L. Pike.* Ed. by Mary R. Wise, Thomas N. Headland, and Ruth M. Brend. Dallas: SIL International and the University of Texas at Arlington, 265–281.

Comrie, Bernard (2013). “Alignment of Case Marking of Full Noun Phrases.” In: *The World Atlas of Language Structures Online.* Ed. by Matthew S. Dryer and Martin Haspelmath. Leipzig: Max Planck Institute for Evolutionary Anthropology.

Comrie, Bernard and Maria Polinsky, eds. (1993). *Causatives and transitivity.* Amsterdam; Philadelphia: John Benjamins Publishing.

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A Sag (2005). “Minimal recursion semantics: An introduction.” In: *Research on language and computation* 3, 281–332.

Copley, Bridget and Fabienne Martin, eds. (2014a). *Causation in grammatical structures.* Oxford: Oxford University Press.

Copley, Bridget and Fabienne Martin (2014b). “Introduction.” In: *Causation in grammatical structures.* Ed. by Bridget Copley and Fabienne Martin. Oxford: Oxford University Press, 1–8.

Copley, Bridget and Phillip Wolff (2014). “Theories of causation should inform linguistic theory and vice versa.” In: *Causation in grammatical structures*. Ed. by Bridget Copley and Fabienne Martin. Oxford: Oxford University Press, 11–57.

Cortés Rodríguez, Francisco José and Ricardo Mairal-Usón (2016). “Building an RRG computational grammar.” In: *Onomazein* 34, 86–117.

Crabbé, Benoit (2005). “Représentation informatique de grammaires fortement lexicalisées [Computational representation of highly lexicalized grammars].” PhD thesis. Université Nancy 2.

Crabbé, Benoit, Denys Duchier, Claire Gardent, Joseph Le Roux, and Yannick Parmentier (2013). “XMG: extensible metagrammar.” In: *Computational Linguistics* 39.3, 591–629.

Cramer, Bart and Yi Zhang (2009). “Construction of a German HPSG grammar from a detailed treebank.” In: *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF)*. Ed. by Tracy Holloway King and Marianne Santaholma. Suntec, 37–45.

Curtis, Christian (2019). “A Parametric Approach to Implemented Analyses: Valence-changing Morphology in the LinGO Grammar Matrix.” In: *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*. Ed. by Magda Ševčíková, Zdeněk Žabokrtský, Eleonora Litta, and Marco Passarotti. Prague, 111–120.

Curtis, Christian Michael (2018). “A parametric implementation of valence-changing morphology in the LinGO Grammar Matrix.” MA thesis. University of Washington.

De Busser, Rik (Dec. 3, 2010). “Beyond the causative continuum.” Manuscript.

DeLancey, Scott (1984). “Notes on agentivity and causation.” In: *Studies in Language. International Journal sponsored by the Foundation “Foundations of Language”* 8.2, 181–213.

Diedrichsen, Elke (2011). “The theoretical importance of constructional schemas in RRG.” In: *New perspectives in role and reference grammar*. Ed. by Wataru Nakamura. Newcastle upon Tyne: Cambridge Scholars Publishing, 168–197.

Diedrichsen, Elke (2012). “What you give is what you GET? On reanalysis, semantic extension and functional motivation with the German bekommen-passive construction.” In: *Linguistics* 50.6, 1163–1204.

Dixon, Robert M. W. (2000). “A typology of causatives: form, syntax and meaning.” In: *Changing valency: Case studies in transitivity*. Ed. by Robert M. W. Dixon and Alexandra Y. Aikhenvald. Cambridge: Cambridge University Press, 30–83.

Dixon, Robert M. W. and Alexandra Y. Aikhenvald, eds. (2000a). *Changing valency: Case studies in transitivity*. Cambridge: Cambridge University Press.

Dixon, Robert M. W. and Alexandra Y. Aikhenvald (2000b). “Introduction.” In: *Changing valency: Case studies in transitivity*. Ed. by Robert M. W. Dixon and Alexandra Y. Aikhenvald. Cambridge: Cambridge University Press, 1–29.

Dmitriev, N. K. (2008 [1948]). *Grammatika baškirskogo jazyka [Grammar of Bashkir]*. Moscow: Nauka.

Dowty, David (1991). “Thematic proto-roles and argument selection.” In: *Language* 67.3, 547–619.

Dryer, Matthew S. (1982). “Passive and inversion in Kannada.” In: *Proceedings of the Eighths Annual Meeting of the Berkeley Linguistics Society*. Ed. by Monica Macaulay, Orin D. Gensler, Claudia Brugman, Inese Čivkulis, Amy Dahlstrom, Katherine Krile, and Rob Sturm. Berkeley: University of California, 311–334.

Dryer, Matthew S. and Martin Haspelmath, eds. (2013). *WALS Online*. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: https://wals.info/.

Duchier, Denys, Brunelle Magnana Ekoukou, Yannick Parmentier, Simon Petitjean, and Emmanuel Schang (2012). “Describing morphologically-rich languages using metagrammars: a look at verbs in Ikota.” In: *Proceedings of the “Workshop on Language Technology for Normalisation of Less-Resourced Languages”*. Ed. by Guy De Pauw, Gilles-Maurice de Schryver, Mikel L. Forcada, Kepa Sarasola, Francis M. Tyers, and Peter Waiganjo Wagacha. Istanbul, 55–66.

Dybo, Anna (2017). “New trends in European studies on the Altaic problem.” In: *Journal of Language Relationship* 14.1-2, 71–106.

Eberhard, David M., Gary F. Simons, and Charles D. Fennig, eds. (2021). *Ethnologue: Languages of the World*. 24th ed. Dallas: SIL International. URL: http://www.ethnologue.com.

Fillmore, Charles J. (1967). “The case for case.” In: *Universals in linguistic theory*. Ed. by E. Bach and R. Harms. New York: Holt, Rinehart & Winston, 1–88.

Fokkens, Antske (2011). “Metagrammar engineering: Towards systematic exploration of implemented grammars.” In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, 1066–1076.

Fokkens, Antske, Tania Avgustinova, and Yi Zhang (2012). “CLIMB grammars: three projects using metagrammar engineering.” In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*. Ed. by Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Istanbul, 1672–1679.

Foley, William and Robert D. Van Valin Jr. (1984). *Functional syntax and universal grammar*. Cambridge: Cambridge University Press.

Foley, William A. (2010). “Clause linkage and nexus in Papuan languages.” In: *Clause Linking and Clause Hierarchy: Syntax and Pragmatics*. Ed. by Isabelle Bril. Amsterdam; Philadelphia: John Benjamins, 27–50.

Gardent, Claire and Laura Kallmeyer (2003). "Semantic construction in feature-based TAG." In: *10th meeting of the European Chapter of the Association for Computational Linguistics (EACL)*. Ed. by Ann Copestake and Jan Hajič. Budapest, 123–130.

Generalova, Valeria (2014). "Differencial'noe markirovanie subjekta v konstrukcijax s sentencial'nymi aktantami [Differencial subject marking in Bashkir constructions with sentential agents]." In: *Materialy XIV Vserossijskoj nauchnoj konferencii «Aktual'nye problemy dialektologii jazykov narodov Rossii» [Proceedings of the 14th Conference «Actual Problems of Dialectology of Minor Languages in Russia»]*. Ufa, 130–136.

Generalova, Valeria (2016). "Vybor strategii oformlenia sentencial'nyx aktantov v baškirskom jazyke [Strategies of complement clauses marking in Bashkir]." In: *Acta Linguistica Petropolitana* 12.1, 323–335.

Generalova, Valeria (Aug. 2019a). *An RRG analysis of Halkomelem ditransitive constructions: Integrating head-marking and dependent-marking properties*. Talk at the International Conference on Role and Reference Grammar 2019. Buffalo, NY, USA.

Generalova, Valeria (Oct. 2019b). *An RRG approach to case assignment in Altaic causative constructions*. Talk at The International Workshop "The syntax and semantics of internal arguments". Bucharest, Romania.

Generalova, Valeria (May 2019c). *Argument linking in some Turkic causative constructions*. Talk at the TreeGraSP Meeting #3. Heinrich Heine University of Dusseldorf. URL: https://treegrasp.phil.hhu.de/talks-and-events/treegrasp-meeting-3/.

Generalova, Valeria (Nov. 2020a). *A metagrammar of three-argument constructions with morphological causatives: general design and first implementations*. Talk at the TreeGraSP Meeting #5. Heinrich Heine University of Dusseldorf. URL: https://treegrasp.phil.hhu.de/talks-and-events/treegrasp-meeting-5/.

Generalova, Valeria (Feb. 2020b). *A verbal alternation construction without macrorole change: Comparing approaches to linking in RRG*. Poster presentation at the 21st Workshop on the Roots of Pragmasemantics. Szklarska Poręba, Poland. URL: https://sites.google.com/view/21st-szklarska-porba-workshop/programme.

Generalova, Valeria (Aug. 2020c). *Applicatives and transitivizers as valence-increasing or valence-maintaining devices in Salish*. Talk at the Workshop "Neglected syntactic functions and non-syntactic functions of Applicative Morphology", SLE Annual Meeting 2020. Online. URL: https://osf.io/fbm5r/.

Generalova, Valeria (Oct. 2020d). *Modeling the variety of constructions with morphological causatives metagrammatically: A first sketch*. Talk at the 10th conference ''Typology of Morphosyntactic Parameters''. Moscow, Russia (online). URL: http://tipl.philol.msu.ru/index.php/science/conferences/TMP2020/programme-eng.

Generalova, Valeria (July 2020e). *Towards a Cross-Linguistic Description of Morphological Causatives: Issues in Syntax-Semantics Linking*. Talk at the WeSSLLI + ESSLLI Virtual Student Session. Online. URL: https://osf.io/dn2j8.

Generalova, Valeria (2020f). "Towards a cross-linguistic description of morphological causatives: issues in syntax-semantics linking." In: *Proceedings of the ESSLLI & WeSSLLI Student Session 2020*. Ed. by Alexandra Pavlova. Brandies University, 55–66. URL: https://www.academia.edu/43943952/Towards_a_cross_linguistic_description_of_morphological_causatives_issues_in_syntax_semantics_linking.

Generalova, Valeria (2020g). "Towards an RRG analysis of constructions with a morphological causative in some Altaic languages." In: *Bucharest Working Papers in Linguistics* XXII.2: *Special issue: The Syntax and Semantics of Internal Arguments*. Ed. by Alina Tigău, 97–120.

Generalova, Valeria (Nov. 2020h). *Yet another attempt at a syntax-driven typology of morphological causative constructions*. Talk at the 17th Conference on Typology and Grammar for Young Scholars. St Petersburg, Russia (online). URL: https://www.youngconfspb.com/e/17th-conference-2020.

Generalova, Valeria (2021a). "A conceptualization of language-specific metagrammatical classes in a multilingual project." In: *Proceedings of the ESSLLI Student Session 2021*. Ed. by Mina Young Pedersen and Alexandra Pavlova, 188–197.

Generalova, Valeria (2021b). "Describing valence-increasing constructions with XMG." In: *Proceedings of the Third International Workshop on Resources and Tools for Derivational Morphology (DeriMo 2021)*. Ed. by Fiammeta Namer, Nabil Hathout, Stéphanie Lignon, Magda Ševčíková, and Zdeněk Žabokrtský. Université de Lorraine, 61–69.

Generalova, Valeria (Aug. 2021c). *Whipping the Linking Algorithm into the feature structure shape*. Talk at The 16th International Conference on Role and Reference Grammar. Toronto, Canada (online).

Generalova, Valeria and Simon Petitjean (2020). "A prototype of a metagrammar describing three-argument constructions with a morphological causative." In: *Typology of Morphosyntactic Parameters* 3.2, 29–51.

Generalova, Valeria and Simon Petitjean (2022). *A metagrammar of causatives in morphologically rich languages*. Ed. by Jernej Vičič and Branko Kavšek.

Grimshaw, Jane (1990). *Argument Structure*. Cambridge, MA: MIT Press.

Gruzdeva, Ekaterina (1998). *Nivkh*. Vol. 111. Languages of the World / Materials. München; Newcastle: Lincom.

Guerrero, Lilián (2007). "Yaqui causation, its form-function interface." In: *Studies in voice and transitivity*. Ed. by Zarina Estrada Fernández, Søren Wichmann, Claudine Chamoreau, and Albert Álvarez González. Munich: Lincom, 201–221.

Guerrero, Lilián and Robert D. Van Valin Jr. (2004). “Yaqui and the analysis of primary object languages.” In: *International Journal of American Linguistics* 70.3, 290–319.

Guirardello, Raquel (1999). “A reference grammar of Trumai.” PhD thesis. Houston, Texas: Rice University.

Guirardello-Damian, Raquel (2003). “Classes verbais e mudanças de valência em Trumai [Verb classes and valence alternations in Trumai].” In: *Proceedings of the Workshop Ergatividade na Amazônia II*. Ed. by Francesc Queixalós. Paris: Centre d’études des langues indigènes d’Amérique, Laboratório de Línguas Indígenas, 195–214.

Guirardello-Damian, Raquel (2004). “Coreference in Trumai.” In: *Proceedings of the Workshop Ergatividade na Amazônia III*. Ed. by Francesc Queixalós. Paris: Centre d’études des langues indigènes d’Amérique, Laboratório de Línguas Indígenas.

Hammarström, Harald, Robert Forkel, Martin Haspelmath, and Sebastian Bank (2022). *Glottolog 4.7*. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: http://glottolog.org.

Haspelmath, Martin (1993). “More on the typology of inchoative/causative verb alternations.” In: *Causatives and transitivity*. Ed. by Bernard Comrie and Maria Polinsky. Amsterdam; Philadelphia: John Benjamins Publishing, 87–121.

Haspelmath, Martin (2005). “Argument marking in ditransitive alignment types.” In: *Linguistic discovery* 3.1, 1–21.

Haspelmath, Martin (2010). “Comparative concepts and descriptive categories in crosslinguistic studies.” In: *Language* 86.3, 663–687.

Haspelmath, Martin (2013). “Ditransitive Constructions: The Verb ‘Give’.” In: *The World Atlas of Language Structures Online*. Ed. by Matthew S. Dryer and Martin Haspelmath. Leipzig: Max Planck Institute for Evolutionary Anthropology. URL: https://wals.info/chapter/105.

Haspelmath, Martin (2014). *How different are head-marking constructions?* Blogpost on “Diversity Linguistics Comment” https://dlc.hypotheses.org/682. Accessed: 2023-07-01.

Haspelmath, Martin and Thomas Müller-Bardey (2004). “Valency change.” In: *Morphology: A handbook on inflection and word formation*. Ed. by Booij Geert, C. Lehmann, Mudgan J., and Skopeteas S. Vol. 2. Berlin; New York: De Gruyter, 1130–1145.

Hemmings, Charlotte (2013). “Causatives and applicatives: The case for polysemy in Javanese.” In: *SOAS Working Papers in Linguistics* 16, 167–194.

Hoffman, Mika Christine (1991). “The syntax of argument-structure-changing morphology.” PhD thesis. Massachusetts Institute of Technology.

Holisky, Dee Ann (1987). “The case of the intransitive subject in Tsova-Tush (Batsbi).” In: *Lingua* 71.1-4, 103–132.

Hopper, Paul J. and Sandra A. Thompson (1980). “Transitivity in grammar and discourse.” In: *Language* 56.2, 251–299.

Horvath, Julia and Tal Siloni (2011). “Causatives across components.” In: *Natural Language & Linguistic Theory* 29.3, 657–704.

Hyman, Larry M. and Karl Zimmer (1976). “Embedded topic in French.” In: *Subject and topic*. Ed. by Charles N. Li. New York: Academic Press, 189–212.

Jakobson, Roman (1936). “Beitrage zur allgemeinen Kasuslehre: Gesamtbedeutungen der russischen Kasus [Contributions to the general theory of case: General meanings of the Russian cases].” In: *Travaux du Cercle Linguistique de Prague*. Vol. 6. Prague. Reprinted in Linda Waugh and Halle, Morris (eds.) (1984). Russian and Slavic grammar: Studies 1931-1981, 59-103. Berlin: Mouton.

Jerro, Kyle (2015). “Revisiting object symmetry in Bantu.” In: *Selected Proceedings of the 44th Annual Conference on African Linguistics*. Ed. by Ruth Kramer, Elizabeth C. Zsiga, and One Tlale Boyer. Somerville, MA: Cascadilla Press, 130–145.

Jerro, Kyle (2017). “The causative–instrumental syncretism.” In: *Journal of Linguistics* 53.4, 751–788.

Jerro, Kyle (2019). “Ingestive verbs, causatives, and object symmetry in Lubukusu.” In: *Linguistic Inquiry* 50.1, 219–232.

Jerro, Kyle (2021). “Applied Objects and the Syntax–Semantics Interface.” In: *Journal of Linguistics* 57.2, 365–403.

Joshi, Aravind K. and Yves Schabes (1997). “Tree-adjoining grammars.” In: *Handbook of Formal Languages. Volume 3: Beyond Words*. Ed. by Grzegorz Rozenberg and Arto Salomaa. Berlin; Heidelberg: Springer, 69–123.

Kailuweit, Rolf (2013). “Radical Role and Reference Grammar (RRRG).” In: *Linking Constructions into Functional Linguistics: The role of constructions in grammar*. Ed. by Brian Nolan and Elke Diedrichsen. Amsterdam; Philadelphia: John Benjamins, 103–142.

Kallmeyer, Laura (2021). “Idea: wrapping for head-marking.” A handout stored on a private cloud storage. June, 17, 2021.

Kallmeyer, Laura, Timm Lichte, Wolfgang Maier, Yannick Parmentier, Johannes Dellert, and Kilian Evang (2008). “TuLiPA: towards a multi-formalism parsing environment for grammar engineering.” In: *Proceedings of the Workshop on Grammar Engineering Across Frameworks*. Ed. by Stephen Clark and Tracy Holloway King. Manchester, 1–8.

Kallmeyer, Laura, Timm Lichte, Rainer Oswald, and Simon Petitjean (2016). “Argument linking in LTAG: A constraint-based implementation with XMG.” In: *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 12)*. Ed. by David Chiang and Alexander Koller. Düsseldorf, 48–57.

Kallmeyer, Laura and Rainer Osswald (2012). “A frame-based semantics of the dative alternation in Lexicalized Tree Adjoining Grammars.” In: *Empirical issues in syntax and semantics* 9, 167–184.

Kallmeyer, Laura and Rainer Osswald (2013). "Syntax-driven semantic frame composition in lexicalized tree adjoining grammars." In: *Journal of Language Modelling* 1.2, 267–330.

Kallmeyer, Laura and Rainer Osswald (2023). "Formalization of RRG syntax." In: *The Cambridge Handbook of Role and Reference Grammar*. Ed. by Delia Bentley, Ricardo Mairal Usón, Wataru Nakamura, and Robert D. Van Valin Jr. Cambridge: Cambridge University Press, 737–784.

Kallmeyer, Laura, Rainer Osswald, and Robert D. Van Valin Jr. (2013). "Tree wrapping for role and reference grammar." In: *Formal Grammar*. Ed. by Glyn Morrill and Mark-Jan Nederhof, 175–190.

Kamali, Beste (2015). "Caseless direct objects in Turkish revisited." In: *ZAS Papers in Linguistics* 58, 107–123.

Kaneko, Tohru (2006). "Note on CAUSE in Nivkh." In: *Journal of Chiba University Eurasian Society* 9, 1–30.

Karlsson, Fred (1999). *Finnish: An Essential Grammar*. London; New York: Routledge.

Kemmer, Suzanne and Arie Verhagen (1994). "The grammar of causatives and the conceptual structure of events." In: *Cognitive linguistics* 5.2, 115–156.

Khelil, Cherifa Ben, Chiraz Ben Othmane Zribi, Denys Duchier, and Yannick Parmentier (2018). "Building a syntactic-semantic interface for a semi-automatically generated TAG for Arabic." In: *The International Arab Journal of Information Technology* 15.3A, 540–549.

Khelil, Chérifa Ben, Denys Duchier, Yannick Parmentier, Chiraz Zribi, and Fériel Ben Fraj (2016). "ArabTAG: from a Handcrafted to a Semi-automatically Generated TAG." In: *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 12)*. Ed. by David Chiang and Alexander Koller. Düsseldorf, 18–26.

Kibrik, Andrej A. (1987). "Mexanizmy ustranenija referencial'nogo konflikta [Mechanisms of referential conflict removal]." In: *Modelirovanie jazykovoj dejatel'nosti v intellektual'nyx sistemax*. Ed. by Aleksandr E. Kibrik and Aleksandr S. Narin'jani. Moscow: Nauka, 128–145.

Kibrik, Andrej A. (1991). "Maintenance of reference in sentence and discourse." In: *Language Typology 1988: Typological Models in the Service of Reconstruction*. Ed. by Winfred P. Lehman and Helen-Jo Jakusz Hewitt. Amsterdam; Philadelphia: John Benjamins, 57.

Kihara, Claudius Patrick (2017). "Aspects of Gĩkũyũ (Kikuyu) Complex Sentences: A Role and Reference Grammar Analysis." PhD thesis. Heinrich-Heine University Düsseldorf.

Kinyon, Alexandra, Owen Rambow, Tatjana Scheffler, SinWon Yoon, and Aravind Joshi (2006). "The metagrammar goes multilingual: A cross-linguistic look at the V2-phenomenon."

In: *Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms*. Ed. by Tilman Becker and Laura Kallmeyer. Sydney, 17–24.

Kiparsky, Paul (2001). “Structural case in Finnish.” In: *Lingua* 111.4-7, 315–376.

Kozinsky, Isaac and Maria Polinsky (1993). “Causee and patient in the causative of transitive: Coding conflict or doubling of grammatical relations.” In: *Causatives and transitivity*. Ed. by Bernard Comrie and Maria Polinsky. Amsterdam; Philadelphia: John Benjamins, 177–240.

Krejnovič, E. A. (1934). “Nivkhskij (gil’ackij) jazyk [Nivkh (Gilyak) language].” In: *Jazyki i pis’mennost’ narodov Severa [Languages and writing of the indigenous peoples of the North]*. Ed. by E. A. Krejnovič. Vol. 3. Leningrad, 181–222.

Krishnamurti, Bhadriraju (2003). *The Dravidian Languages*. Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511486876.

Kulikov, Leonid (1998). “Causative constructions in Tuvinian: towards a typology of transitivity.” In: *The Mainz Meeting: Proceedings of the Seventh International Conference on Turkic Linguistics*. Ed. by Lars Johanson et al., 258–264.

Kulikov, Leonid (2001). “Causatives.” In: *Language Typology and Language Universals / Sprachtypologie und sprachliche Universalien / La typologie des languages et les universaux lin-guistiques. An International Handbook / Ein internationales Handbuch / Manual international*. Ed. by Martin Haspelmath, Ekkerhard König, Wulf Oesterreicher, and Wolfgang Raible. Vol. 2. Berlin; New York: De Gruyter, 886–898.

Kumaxov, M. A., ed. (2006). *Kabardino-Čerkesskij jazyk [Kabardian language]*. Vol. 1. Nal’čik: El’-Fa.

Lakoff, George (1977). “Linguistic gestalts.” In: *Proceedings of the Thirteenth Annual Meeting of the Chicago Linguistic Society*. Ed. by Woodford E. Beach, Samuel E. Fox, and Shulamith Philosoph. Chaicago: Chicago Linguistic Society, 236–287.

Lakoff, George (1982). *Categories and cognitive models*. Cognitive Science Reports 2. Berkeley: University of California, Cognitive Science Program.

Lakoff, George (1984). *Classifiers as a reflection of mind: a cognitive model approach to prototype theory*. Cognitive Science Reports 19. Berkeley: University of California, Cognitive Science Program.

Lakoff, George (2007). “Cognitive models and prototype theory.” In: *The cognitive linguistics reader*. Ed. by Vyvyan Evans, Benjamin K. Bergen, and Jörg Zinken. London: Equinox, 130–167.

Larousse.fr. Dictionnaire de français (Accessed: 2026-05-01). https://www.larousse.fr/dictionnaires/francais-monolingue. Paris.

Latrouite, Anja (2014). “Event-structural prominence and forces in verb meaning shifts.” In: *Causation in grammatical structures*. Ed. by Bridget Copley and Fabienne Martin. Oxford: Oxford University Press, 372–294.

Letuchiy, Alexander (2006). “Case marking, possession and syntactic hierarchies in Khakas causative constructions in comparison with other Turkic languages.” In: *Case, Valency and Transitivity*. Ed. by L. Kulikov, A. Malchukov, and P. de Swart. Amsterdam; Philadelphia: John Benjamins, 417–439.

Letuchiy, Alexander B. (2012). “Ergativity in the Adyghe system of valency-changing derivations.” In: *Ergativity, valency and voice*. Ed. by Gilles Authier and Katharina Haude. Berlin; Boston: De Gruyter Mouton, 323–354.

Li, Chao (2008). “On the argument realization of causatives.” In: *LACUS Forum 34: Speech and Beyond*. Ed. by Patricia Sutcliffe, Lois M. Stanford, and Arle R. Lommel. Houston, TX: LACUS.

Li, Chao (2020). “Causer and causee as two higher-ranked thematic roles.” In: *Language and Linguistics* 21.4, 601–635.

Li, Yafei (1990). “X^0-Binding and Verb Incorporation.” In: *Linguistic Inquiry* 21.3, 399–426.

Lichte, Timm and Simon Petitjean (2015). “Implementing semantic frames as typed feature structures with XMG.” In: *Journal of Language Modelling* 3.1, 185–228.

Luraghi, Silvia and Elisa Roma (2021). *Valency over time: Diachronic perspectives on valency patterns and valency orientation*. Berlin; Boston: De Gruyter.

Lyutikova, Ekaterina and Sergei Tatevosov (2014). “Causativization and event structure.” In: *Causation in grammatical structures*. Ed. by Bridget Copley and Fabienne Martin. Oxford: Oxford University Press, 279–327.

Magnana Ekoukou, Brunelle (July 2014). “PFM: Ikota’s morphology implementation in XMG (PFM : pour une implémentation de la morphologie de l’ikota dans XMG) [in French].” In: *TALN-RECITAL 2014 Workshop TALAf 2014 : Traitement Automatique des Langues Africaines (TALAf 2014: African Language Processing)*. Marseille, France: Association pour le Traitement Automatique des Langues, 134–140. URL: https://aclanthology.org/W14-6504.

Maho, Jouni Filip, ed. (2009). *NUGL Online. The online version of the New Updated Guthrie List, a referential classification of the Bantu languages.*

Malihah, Noor (2014). “Standard Javanese Causatives in online editorials and short stories.” In: *Register Journal* 7.1, 51–74.

Manninen, Satu and Diane Nelson (2004). “What is a passive? The case of Finnish.” In: *Studia linguistica* 58.3, 212–251.

Marantz, Alec P. (1984). *On the Nature of Grammatical Relations*. Cambridge, MA: MIT Press.

Marantz, Alec P. (1985). “Lexical decomposition vs. affixes as syntactic constituents.” In: *Papers from the parasession on Causatives and Agentivity at CLS 21*. Chicago: Chicago Linguistic Society, 154–171.

Marelj, Marijana (2004). “Middles and argument structure across languages.” PhD thesis. University of Utrecht.

Margetts, Anna (2007). “Learning verbs without boots and straps? The problem of ‘give’ in Saliba.” In: *Cross-linguistic perspectives on argument structure: Implications for learnability*. Ed. by Melissa Bowerman and Penelope Brown. London; New York: Routledge, 111–141.

Margetts, Anna and Peter K. Austin (2007). “Three-participant events in the languages of the world: Towards a crosslinguistic typology.” In: *Linguistics* 45.3.

Martin, Fabienne and Florian Schäfer (2014). “Causation at the syntax-semantics interface.” In: *Causation in grammatical structures*. Ed. by Bridget Copley and Fabienne Martin. Oxford: Oxford University Press, 209–244.

Matasović, Ranko (2009). “A new typology of control constructions within the framework of Role and Reference Grammar.” In: *Studies in Role and Reference Grammar*. Ed. by Sergio Ibáñez Valenzuela Lilián Guerrero and Valeria Belloro. México City: Universidad Nacional Autónoma de México, 305–318.

Matasović, Ranko (2010a). *A short grammar of East Circassian (Kabardian). Ninth version.* Zagreb. Unpublished manuscript.

Matasović, Ranko (2010b). “Kabardian causatives, reflexives, and case marking domains.” In: *Suvremena lingvistika* 36.69, 45–64.

Matteson, Esther (1965). *The Piro (Arawakan) language.* Berkeley: University of California Press.

Monod Becquelin, Aurore and Cédric Becquey (2012). “Case patterns and verb classes in Trumai.” In: *Ergativity, valency and voice.* Ed. by Gilles Authier and Katharina Haude. Berlin; Boston: De Gruyter Mouton, 289–322.

Monod Becquelin, Aurore and Cédric Becquey (2013). “Transitivité de l’action et ses expressions en Trumai (langue isolée du Haut Xingú, Brésil) [The transitivity of action and its expressions in Trumai (an isolate language of Upper Xingú, Brasil].” In: *Ateliers d’anthropologie* 39. DOI: https://doi.org/10.4000/ateliers.9478.

Nadathur, Prerna and Sven Lauer (2020). “Causal necessity, causal sufficiency, and the implications of causative verbs.” In: *Glossa* 5.1, 1–37.

Nakamura, Wataru (2011). “Case syncretism in typological perspective: an RRG-OT account.” In: *New Perspectives in Role and Reference Grammar.* Ed. by Wataru Nakamura. Newcastle upon Tyne: Cambridge Scholars Publishing, 35–64.

Nakamura, Wataru (2021). “A Neo-Jakobsonian Account of Default Oblique Cases: Instrumental vs. Dative.” In: *Challenges at the Syntax-Semantics-Pragmatics Interface: A Role and Reference Grammar Perspective.* Ed. by Robert D. Van Valin Jr. Newcastle upon Tyne: Cambridge Scholars Publishing, 135–168.

Nedjalkov, V. P. and G. A. Otaina (2012). "Sintaksis nivkhskogo jazyka [The Syntax of Nivkh]." In: *Ocherki po sintaksisu nivkhskogo jayzka [Essays in Nivkh Syntax]*. Ed. by E. V. Geniušene and S. Yu. Dmitrenko. Moscow: Znak, 18–237.

Nedyalkov, V. P., G. A. Otaina, and A. A. Xolodovič (1969). "Morfologičeskiy i leksičeskiy kauzativy v nivxskom jazyke [Morphological and lexical causatives in Nivkh]." In: *Typology of causative constructions: Morphological causative*. Ed. by A. A. Xolodovič. Leningrad: Nauka, 179–199.

Nedyalkov, V. P. and G. G. Sil'nitskiy (1969a). "Tipologija kauzativnyx konstrukcij [Typology of causative constructions]." In: *Typology of causative constructions: Morphological causative*. Ed. by A. A. Xolodovič. Leningrad: Nauka, 5–19.

Nedyalkov, V. P. and G. G. Sil'nitskiy (1969b). "Tipologija morfologičeskogo i leksičeskogo kauzativov [Typology of morphological and lexical causatives]." In: *Typology of causative constructions: Morphological causative*. Ed. by A. A. Xolodovič. Leningrad: Nauka, 20–50.

Nelson, Diane (1999). "Events, arguments, and causative psych predicates in Finnish." In: *Leeds Working Papers in Linguistics and Phonetics* 7, 145–171.

Newman, John (1998). *The linguistics of giving*. Amsterdam; Philadelphia: John Benjamins.

Nieuwland, Mante S. and Jos J. A. Van Berkum (2008). "The neurocognition of referential ambiguity in language comprehension." In: *Language and Linguistics Compass* 2.4, 603–630.

Nikanne, Urpo (1989). "On locative cases in Finnish." In: *Papers from the Eleventh Scandinavian Conference of Linguistics*. Ed. by Jussi Niemi. Vol. 1. Joensuu: University of Joensuu, Faculty of Arts, 147–164.

Nikanne, Urpo (1997). "Locative case adjuncts in Finnish: Notes on syntactico-semantic interface." In: *Nordic journal of linguistics* 20.2, 155–178.

Nikanne, Urpo (2015). "On assigning semantic cases in Finnish." In: *Case and other functional categories in Finnish syntax*. Ed. by Anders Holmberg and Urpo Nikanne. Berlin; Boston: De Gruyter Mouton, 75–88.

Nolan, Brian (2011). "Meaning construction and grammatical inflection in the layered structure of the Irish word: an RRG account of morphological constructions." In: *New perspectives in role and reference grammar*. Ed. by Wataru Nakamura. Newcastle upon Tyne: Cambridge Scholars Publishing, 64–103.

Nolan, Brian and Elke Diedrichsen, eds. (2013). *Linking Constructions into Functional Linguistics: The role of constructions in grammar*. Amsterdam; Philadelphia: John Benjamins.

Nolan, Brian and Elke Diedrichsen, eds. (2019). *Linguistic Perspectives on the Construction of Meaning and Knowledge*. Newcastle upon Tyne: Cambridge Scholars Publishing.

Nolan, Brian, Gudrun Rawoens, and Elke Diedrichsen, eds. (2015). *Causation, Permission, and Transfer: Argument realisation in GET, TAKE, PUT, GIVE and LET verbs*. Amsterdam; Philadelphia: John Benjamins.

Nolan, Brian and Y. Hensman Salem (2010). "UniArab: An RRG Arabic-to-English Machine Translation Software." In: *Proceedings of the 10th International RRG Conference (RRG 2009)*. Ed. by Wataru Nakamura. Berkeley: University of California, 243–270.

Nurhayani, Ika (2014). "A unified account of the syntax of valence in Javanese." PhD thesis. Cornell University.

Nurse, Derek and Gérard Philippson (2003). *The Bantu languages*. London: Routledge.

Osswald, Rainer and Laura Kallmeyer (2018). "Towards a formalization of Role and Reference Grammar." In: *Applying and expanding Role and Reference Grammar*. Ed. by Rolf Kailuweit, Eva Staudinger, and Lisann Künkel. Freiburg: Albert-Ludwigs-Universität, 355–378.

Ovsjannikova, Maria, Sergey Say, Ekaterina Aplonova, Anna Smetina, and Elena Sokur (2019). *Spoken corpus of Bashkir language of Raxmetovo and Baimovo villages*. Saint Petersburg and Moscow. URL: http://lingconlab.ru/spoken_bashkir/.

Öztürk, Balkız (2005). *Case, referentiality and phrase structure*. Amsterdam; Philadelphia: John Benjamins.

Panfilov, V. Z. (1962). *Grammatika nivkhskogo jayzka [The Grammar of Nivkh]*. Vol. 1. Moscow; Leningrad: Nauka.

Parsons, Terence (1990). *Events in the semantics of English: A study in subatomic semantics*. Cambridge, MA: MIT Press.

Perekhval'skaya, E. V. (2017). "Kauzativnye konstrucii v baškirskom jazyke [Causative constructions in Bashkir]." In: *Acta Linguistica Petropolitana* 13.1, 231–254.

Pereltsvaig, Asya (2012). "Introduction." In: *Languages of the World: An Introduction*. Cambridge: Cambridge University Press, 1–12.

Periñán-Pascual, Carlos (2013). "Towards a model of constructional meaning for natural language understanding." In: *Linking Constructions into Functional Linguistics: The role of constructions in grammar*. Ed. by Brian Nolan and Elke Diedrichsen. Amsterdam; Philadelphia: John Benjamins, 103–142.

Periñán-Pascual, Carlos and Francisco Arcas-Túnez (2014). "The implementation of the CLS constructor in ARTEMIS." In: *Language Processing and Grammars*. Ed. by Carlos Periñán-Pascual and Brian Nolan. Amsterdam; Philadelphia: John Benjamins, 165–196.

Perrier, Guy (2000). "Interaction grammars." In: *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*. Saarbrücken, 600–606.

Petitjean, Simon, Denys Duchier, and Yannick Parmentier (2016). "XMG 2: Describing Description Languages." In: *Logical Aspects of Computational Linguistics. Celebrating*

20 Years of LACL (1996–2016). Ed. by Maxime Amblard, Philippe de Groote, Sylvain Pogodalla, and Christian Retoré. Berlin; Heidelberg: Springer, 255–272.

Podlesskaya, Vera I. (1993). “Causatives and causality: towards a semantic typology of causal relations.” In: *Causatives and transitivity.* Ed. by Bernard Comrie and Maria Polinsky. Amsterdam; Philadelphia: John Benjamins, 165–176.

Pylkkänen, Liina (1997). “The linking of event structure and grammatical functions in Finnish.” In: *Proceedings of the LFG97 Conference.* Ed. by Miriam Butt and Tracy Holloway King. Standford, CA: CSLI Publications, 1–15.

Pylkkänen, Liina (2000). “Representing causatives.” In: *Semantics and Linguistic Theory* 10, 132–148.

Pylkkänen, Liina (2008). *Introducing arguments.* Cambridge, MA: MIT Press.

Pylkkänen, Mariliina (2002). “Introducing arguments.” PhD thesis. Massachusetts Institute of Technology.

Qomariana, Yana (2016). “Causative Constructions in Javanese.” In: *Lingual: Journal of Language and Culture* 1.1, 2–7.

Rappaport Hovav, Malka and Beth Levin (1998). “Building Verb Meanings.” In: *The Projection of Arguments: Lexical and Compositional Factors.* Ed. by Miriam Butt and Wilhelm Geuder. Stanford, CA: CSLI Publications, 97–134.

Rassadin, Valentin (2019). *Ocherki po istorii slozhenija tjurko-mongol’skoj jazykovoj obshhnosti [Essays on the history of Turkic-Mongolic language community].* St Petersburg: Nestor-Istorija.

Reinhart, Tanya (2002). “The theta system: An overview.” In: *Theoretical Linguistics* 28, 229–290.

Robbeets, Martine (2015). *Diachrony of Verb Morphology: Japanese and the Transeurasian Languages.* Berlin; München; Boston: De Gruyter Mouton.

Roberts, Linda (1995). “Pivots, voice and macroroles: from Germanic to Universal grammar.” In: *Australian Journal of Linguistics* 15.2, 157–214.

Rosch, Eleanor (1978). “Principles of categorization.” In: *Principles of categorization.* Ed. by Eleanor Rosch and Barbara B Lloyd. Hillsdale, NJ: Lawrence Erlbaum, 27–48.

Rosen, Sara Thomas (1990). “Argument structure and complex predicates.” PhD thesis. Brandeis University.

Saksena, Anuradha (1980). “The affected agent.” In: *Language* 56.4, 812–826.

Savary, Agata, Simon Petitjean, Timm Lichte, Laura Kallmeyer, and Jakub Waszczuk (2020). “Lexical encoding of multiword expressions in XMG.” In: *Actes des 2èmes journées scientifiquesdu Groupement de Recherche Linguistique Informatique Formelle et de Terrain (LIFT).* Ed. by Thierry Poibeau, Yannick Parmentier, and Emmanuel Schang, 59–62.

Say, Sergey (2009). “Argumentnaja struktura kalmyckix kauzativnyx konstrukcij [Argument structure of Kalmyk causative constructions].” In: *Acta Linguistica Petropolitana* 2, 387–464.

Schachter, Paul (1977). “Reference-related and role-related properties of subjects.” In: *Grammatical relations*. Ed. by Peter Cole and Jerrold M. Sadock. Leiden: Brill, 279–306.

Seyffarth, Esther (2019). “Modeling the induced action alternation and the caused-motion construction with Tree Adjoining Grammar (TAG) and semantic frames.” In: *Proceedings of the IWCS 2019 Workshop on Computing Semantics with Types, Frames and Related Structures*. Ed. by Rainer Osswald, Christian Retoré, and Peter Sutton. Gothenburg: Association for Computational Linguistics, 19–27.

Shibatani, Masayoshi, ed. (1976a). *Syntax and semantics: The grammar of causative constructions*. New York: Academic Press.

Shibatani, Masayoshi (1976b). “The Grammar of Causative Constructions: A Conspectus.” In: *The grammar of causative constructions*. Ed. by Masayoshi Shibatani. New York: Academic Press, 1–10.

Shibatani, Masayoshi and Prashant Pardeshi (2002). “The causative continuum.” In: *The Grammar of Causation and Interpersonal Manipulation*. Ed. by Masayoshi Shibatani, 85–126.

Sikuku, Justine M., Michael Diercks, and Michael R. Marlo (2018). “Pragmatic effects of clitic doubling: Two kinds of object markers in Lubukusu.” In: *Linguistic Variation* 18.2, 359–429.

Skirgård, Hedvig et al. (Apr. 2023). *Grambank v1.0*. Version 1.0.3. Zenodo. DOI: 10.5281/zenodo.7844558.

Smeets, Rieks (1992). “On valencies, actants and actant coding in Circassian.” In: *Caucasian perspectives*. Ed. by George Hewitt. München: Lincom, 98–144.

Sofwan, Ahmad (2010). “Applicative Constructions in Javanese.” In: *Language Circle: Journal of Language and Literature* 5.1, 45–70.

Song, Jae Jung (1991). “Causatives and universal grammar: An alternative interpretation.” In: *Transactions of the Philological Society* 89.1, 65–94.

Song, Jae Jung (1996). *Causatives and Causation: A Universal-Typological Perspective*. London; New York: Routledge.

Sridhar, Shikaripur N (1976). “Dative subjects, rule government and relational grammar.” In: *Studies in the Linguistic Sciences* 6.1, 130–151.

Starostin, Sergey (1991). *Altajskaja problema i proisxozdenije japonskogo jazyka [The Altaic problem and origins of Japanese]*. Moscow: Nauka.

Stolz, Thomas, Cornelia Stroh, and Aina Urdze (2013). “Comitatives and Instrumentals (v2020.3).” In: *The World Atlas of Language Structures Online*. Ed. by Matthew S. Dryer and Martin Haspelmath. Zenodo. URL: https://doi.org/10.5281/zenodo.7385533.

Suhandano (1994). "Grammatical relations in Javanese. A short description." MA thesis. Canberra: The Australian National University.

Tamm, Anne and Piibi-Kai Kivik (2017). "A learner's grammar for Estonian: Insights from typological research." In: *The Journal of Estonian and Finno-Ugric Linguistics* 8.1, 267–285.

Toivainen, Jorma (2015). "The nature of the accusative in Finnish." In: *Case and Other Functional Categories in Finnish Syntax*. Ed. by Anders Holmberg and Urpo Nikanne. Berlin; Boston: De Gruyter Mouton, 111–128.

Toratani, Kiyoko (2002). "The morphosyntactic structure and logical structures of compound verbs in Japanese." PhD thesis. State University of New York at Buffalo.

Vainikka, Anne (2015). "The three structural cases in Finnish." In: *Case and Other Functional Categories in Finnish Syntax*. Ed. by Anders Holmberg and Urpo Nikanne. Berlin; Boston: De Gruyter Mouton, 129–160.

Van der Wal, Jenneke (2018). "Flexibility in symmetry: An implicational relation in Bantu double object constructions." In: *Order and structure in syntax II: Subjecthood and argument structure (Open Generative Syntax 2)*. Ed. by Michelle Sheenan and Laura R. Bailey. Berlin: Language Science Press, 115–152.

Van Valin Jr., Robert D. (1977). "Ergativity and the universality of subjects." In: *Papers from the Thirteenth regional meeting of the Chicago Linguistic Society*. Ed. by Woodford E. Beach, Samuel E. Fox, and Shulamith Philosoph. Chicago: Chicago Linguistic Society, 689–705.

Van Valin Jr., Robert D. (1991). "Another look at Icelandic case marking and grammatical relations." In: *Natural Language & Linguistic Theory* 9.1, 145–194.

Van Valin Jr., Robert D. (2004). "Semantic macroroles in role and reference grammar." In: *Semantische Rollen [Semantic roles]*. Ed. by Rolf Kailuweit and Martin Hummel. Tübingen: Gunter Narr Verlag, 62–82.

Van Valin Jr., Robert D. (2005). *Exploring the syntax-semantics interface*. Cambridge: Cambridge University Press.

Van Valin Jr., Robert D. (2006). "Semantic macroroles and language processing." In: *Semantic Role Universals and Argument Linking*. Ed. by Ina Bornkessel, Matthias Schlesewsky, Bernard Comrie, and Angela D. Friederici. Berlin, New York: Mouton de Gruyter, 263–301.

Van Valin Jr., Robert D. (2008). "RPs and the nature of lexical and syntactic categories in Role and Reference Grammar." In: *Investigations of the syntax-semantics-pragmatics interface*, 161–178.

Van Valin Jr., Robert D. (2013). "Head-marking languages and linguistic theory." In: *Language typology and historical contingency: In honor of Johanna Nichols*. Ed. by

Balthasar Bickel, Leonore A. Grenoble, David A. Peterson, and Alan Timberlake. Amsterdam; Philadelphia: John Benjamins, 91–124.

Van Valin Jr., Robert D. (2018a). "Dative case and oblique subjects." In: *Non-Canonically Case-Marked Subjects. The Reykjavík-Eyjafjallajökull papers*. Ed. by Jóhanna Barðdal, Na'ama Pat-El, and Stephen Mark Carey. Amsterdam; Philadelphia: John Benjamins, 115–131.

Van Valin Jr., Robert D. (2018b). "Some issues regarding (active) accomplishments." In: *Applying and expanding Role and Reference Grammar*. Ed. by Rolf Kailuweit, Eva Staudinger, and Lisann Künkel. Freiburg: Albert-Ludwigs-Universität, 71–93.

Van Valin Jr., Robert D. (2021). "Cosubordination." In: *Challenges at the Syntax-Semantics-Pragmatics Interface: A Role and Reference Grammar Perspective*. Ed. by Robert D. Van Valin Jr. Newcastle upon Tyne: Cambridge Scholars Publishing, 241–256.

Van Valin Jr., Robert D. (2023a). "Grammatical Aspects of Language Processing in the Brain: A Role and Reference Grammar Perspective." In: *The Cambridge Handbook of Role and Reference Grammar*. Ed. by Delia Bentley, Ricardo Mairal Usón, Wataru Nakamura, and Robert D. Van Valin Jr. Cambridge: Cambridge University Press, 693–736.

Van Valin Jr., Robert D. (2023b). "Principles of RRG." In: *The Cambridge Handbook of Role and Reference Grammar*. Ed. by Delia Bentley, Ricardo Mairal Usón, Wataru Nakamura, and Robert D. Van Valin Jr. Cambridge: Cambridge University Press, 17–178.

Van Valin Jr., Robert D. and Elke Diedrichsen (2006). *A Bonsai grammar of German*. Manuscript.

Van Valin Jr., Robert D. and Randy J. LaPolla (1997). *Syntax: Structure, meaning, and function*. Cambridge: Cambridge University Press.

Van Valin Jr., Robert D. and David P. Wilkins (1996). "The Case for 'Effector': Case Roles, Agents, and Agency Revisited." In: *Grammatical Constructions. Their Form and Meaning*. Ed. by Masayoshi Shibatani and Sandra A. Thompson. Oxford: Clarendon Press, 289–321.

Vendler, Zeno (1967). *Linguistics in philosophy*. Ithaca: Cornell University Press.

Vihman, Virve-Anneli (2004). "Valency reduction in Estonian." PhD thesis. University of Edinburgh.

Wali, Kashi (1981). "Cause, causer and causee: a semantic perspective." In: *Journal of Linguistics* 17.2, 289–308.

Weir, David J. (1988). "Characterizing Mildly Context-Sensitive Grammar Formalisms." PhD thesis. University of Pennsylvania.

Xia, Fei, Martha Palmer, and K. Vijay-Shanker (2005). "Automatically generating Tree Adjoining Grammars from abstract specifications." In: *Computational Intelligence* 21.3, 246–285.

Xolodovič, A. A. (1969). *Tipologija kauzativnyx konstrukcij: Morfologičeskij kauzativ [Typology of causative constructions: Morphological causative]*. Leningrad: Nauka.

Yuldashev, A. A., ed. (1981). *Grammatika sovremennogo baškirskogo literaturnogo jazyka [Grammar of contemporary literary Bashkir]*. Moscow: Nauka.

Zinova, Yulia (2017). “Modelling Derivational Morphology: A Case of Prefix Stacking in Russian.” In: *Formal Grammar*. Ed. by Annie Foret, Reinhard Muskens, and Sylvain Pogodalla. Berlin; Heidelberg: Springer, 125–141.

Zinova, Yulia (2021). *Russian verbal prefixation: A frame semantic analysis*. Vol. 7. Empirically Oriented Theoretical Morphology and Syntax. Berlin: Language Science Press.

Zinova, Yulia and Laura Kallmeyer (2012). “A frame-based semantics of locative alternation in LTAG.” In: *Proceedings of the 11th international workshop on Tree Adjoining Grammars and Related Formalisms (TAG+ 11)*. Ed. by Giorgio Satta and Chung-Hye Han. Paris, 28–36.

I affirm in lieu of oath that the dissertation has been prepared by me independently and without unauthorized outside assistance, in compliance with the “Regulations on the Principles for Ensuring Good Scientific Practice at Heinrich Heine University”.

V. Generalova

VALERIA GENERALOVA

Theoretical and Computational Linguist

valeria-generalova.com   @ generalova.valeria@gmail.com

EDUCATION

Ph. D. in Linguistics

Heinrich Heine University of Düsseldorf

Jul 2018 – Sep 2023

- Working contract with TreeGraSP (ERC Consolidator Grant)
- Thesis: **Three-argument causative constructions in Role and Reference Grammar: formalization and implementation**
- Supervisor: Laura Kallmeyer, Co-supervisor: Robert Van Valin, Advisor: Rainer Osswald

M. A. in Computational Linguistics (with excellence)

Russian State University for the Humanities (Moscow)

Sep 2016 – Jul 2018

- Thesis: **Automated detection of users' negative feedback in human-machine natural language dialogues** (in Russian)
- Supervisor: Anatoliy Starostin (Yandex), Co-supervisor: Sergey Sharov (RSUH)

Exchange Student (Faculty of Arts)

University of Groningen (Netherlands)

Feb 2017 – Jun 2017

Bachelor in Theoretical and Experimental Linguistics (with excellence)

Saint Petersburg State University

Sep 2012 – Jul 2016

- Thesis: **Reflection of semantic and syntactic features of the source verb in Russian deverbal adjectives** (in Russian)
- Supervisor: Sergey Say

Exchange Student (Faculty of Arts)

University of Lausanne (Switzerlad)

Feb 2014 – Jun 2014

Certificate of Secondary Education with excellence

Gymnasium 155, St Petersburg

Sep 2001 – Jun 2012

LANGUAGES

Russian (native)
English
German (TestDaF TDN 5)
French (DALF C1)
Spanish

ACCOMPLISHEMENTS

Grants and awards

- **Scholarships**
  - Scholarship from St Petersburg State University for an Exchange semester in the University of Lausanne (tuition, monthly allowance, travel)
  - Erasmus Mundus Scholarship for a semester at the University of Groningen (tuition fees, monthly allowance, travel)
- **Travel and Conference grants**
  - 2 × ESSLLI (2016, 2017) Grants (tuition, accomodation, travel)
  - V-NYI 2021 mini-grant (conference fees)
- **Best Paper Awards**
  - Best B. A. Thesis in Philology and Linguistics of the Promotion 2012 in St Petersburg State University

Miscellanea

- Developed a script for preparing a concordance for the dictionary **"Proverbs and sayings of the Petrine time as a cultural phenomenon of language reforms (retrospective and prospects)"** (eds. V. M. Mokienko et al., SPbSU, in press)
- Contributed to the **textbook "Linguistic study of stereotypes"** (Published in St Petersburg, 2015, ed. E. Vilinbakhova) by creating test questions and problems. Available at the SPbSU Research Repository
- Took part in **two linguistic fieldwork trips** (2015, 2014) to study Bashkir language (a joint project of SPbSU and ILS RAS)

ACTIVITIES

Teaching

- **Role and Reference Grammar – Linguistic and Computational Aspects** (together with Rainer Osswald, Laura Kallmeyer, Simon Petitjean), SoSe 2022, Heinrich Heine University of Dusseldorf, Advanced Undergraduate & Master
- **Linguistic Resources**, Heinrich Heine University of Dusseldorf, SoSe, Undergraduate; **2023**: on-site, **2022**: on site, together with Regina Stodden, **2021**: online, together with Regina Stodden
- **Advanced Philology**, 2015-2018, St Petersburg, High School students

Additional Training

- Mentee at the Selma Meyer Mentoring Program (Heinrich Heine University of Dusseldorf, 2021-2022)
- ROLLING Institute of Argument Structure 2021 – 9 × 10-hour courses (online)
- Virtual New York Institute Winter Session 2021 (online)
- European Summer School for Language, Logic and Information (ESSLLI) 2016, 2017, 2018, 2019, 2020
- Late Summer School "Machine Learning for Language Analysis" 2018 (Cologne, Germany)
- EDUFI (CIMO) Winter School 2018 "Big and small data: from data to understanding" (Tvärminne, Finland)
- The ShanghAI Lectures on Natural and Artificial Intelligence 2016 (in partnership with RSUH)