Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Past, present and future: Computational approaches to mapping historical Irish cognate verb forms

Theodorus Leman Franciscus Fransen

Thesis submitted for the Degree of Doctor of Philosophy

Centre for Language & Communication Studies
School of Linguistic, Speech and Communication Sciences
Trinity College Dublin
The University of Dublin

2019

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Summary

This thesis investigates how computational methods can be used to enhance our understanding of the significant historical developments in the verbal system between Old Irish (c. 8th–9th centuries A.D.) and Modern Irish (13th century onwards). Out of all grammatical subsystems, the verbal system is subject to the most severe morphological changes between Old and Modern Irish, i.e., during the Middle Irish period (c. 10th–12th centuries). The main contribution of this thesis is the creation of a morphological Finite-State Transducer (FST) for Old Irish, focusing on verbs, successfully implemented in the finite-state tool `foma` (Hulden 2009). The FST is an important advancement in Natural Language Processing for Old Irish and will assist research in various linguistic subdisciplines as well as in medieval Irish philology.

Chapter 1 demonstrates that a hiatus exists in digital linguistic support for historical Irish language periods. This hiatus in coverage and continuity is particularly true for Early Modern Irish (c. 13th–mid 17th centuries); it not only deters one from carrying out a systematic diachronic analysis of the verbal system, but also complicates the task of satisfactorily linking up available and emerging lexical resources for Old Irish and Modern Irish. The otherwise invaluable electronic Dictionary of the Irish Language (eDIL), covering the period c. 700–1700, was not deemed feasible as a starting point for a morphological parser for Old Irish. However, it is part of a proposed linking framework presented in Chapter 6. The focus in the current work is on Old Irish (rather than Middle or Early Modern Irish) due to its (comparatively) uniform, normative and well-resourced nature.

Chapter 2 focuses on the complex verbal system of Old Irish, introducing in a stepwise fashion the various elements that constitute the verbal complex: one accentual unit which may comprise, apart from the lexical root, various prefixes, infixes and suffixes. The concept of the verbal complex is also of significant importance for the computational implementation. The stress system of Old Irish results in a bewildering array of inflectional variation, especially in the 'middle part' of the verb. A major challenge that had to be overcome in this thesis is capturing the complex interplay between morphology and phonology in order to arrive at a feasible way of implementing Old Irish verb morphology programmatically. The current work focuses on the weak verb classes W1 and W2a, whose inflection patterns are much more predictable than the inflection patterns of the strong types.

Chapter 3 reports on a plethora of techniques used in Natural Language Processing for historical texts. No 'one-size-fits-all' algorithm currently exists. Lemmatisation is often aided by

morphological or orthographical rules, or by approximate matching techniques (string similarity). An approximate matching algorithm is used in Dereza's (2016) Early Irish lemmatiser, based on eDIL. If a Part-Of-Speech Tagger for a modern variety is available, one can develop a standardisation module to bring historical forms in line with a modern standard, which are subsequently input to the tagger. This approach is employed in the context of the Royal Irish Academy's *Corpas Stairiúil na Gaeilge* ('Historical Irish Corpus') 1600–1926.

Due to the linguistic distance between Old and Modern Irish, adapting available NLP tools for the modern language was not an option. It was therefore decided to build a morphological parser for Old Irish from the ground up. Together with the tagging tools for Modern Irish, it is envisaged to constitute the core of a 'two-pronged attack': two morphological/morphosyntactic tools for normative and (relatively) well-resourced language varieties at opposite ends of the (historical) chronological spectrum, with standardisation methods to arrive at either end.

Chapter 4 covers in detail the building of a morphological parser for Old Irish using the finite-state paradigm of two-level morphology. A key aspect of the FST implementation is the encoding of what is called the 'monolithic stem' in this work: a non-derived multi-morpheme base, not trivially segmentable on the surface, reflecting the 'middle part' of the verbal complex—crucial for operating with straightforward prefixation and suffixation rules. Another important aspect of the finite-state implementation includes the creation of two main lexicons: one for proclitics (e.g., *ní* 'not') and one for the stems and endings, accompanied by a set of fine-grained morphotactic and morphophonemic rules.

Chapter 5 is dedicated to testing the FST using a digital edition of the text *Táin Bó Fraích* (TBF), based on Meid (1974). This chapter aims to find out how well the morphological analyser performs, and which issues one might encounter when applying an Early Irish text to the FST for verbs. For the 27 weak verb lemmas (Old Irish W1 and W2a) in this text, which are at the heart of my study, 36 unique inflected forms out of 50 in total (72%) received a morphological parse. After using Dereza's (2016) Lemmatiser, an additional 10 were found to be correctly recognised. After having implemented a selection of function words and personal names, it was found that 9.6% of the total amount of words in TBF were covered; comparing this figure against another four Old Irish narrative texts points to a similar figure of about 10%.

Chapter 6 concludes the work and provides a roadmap for the future. It proposes a preliminary infrastructure for bidirectional mappings between cognate verb forms. The first 'route' involves lexical tag mappings between my morphological FST for Old Irish and the one for Modern Irish (Uí Dhonnchadha & van Genabith 2006). A second strategy is to link eDIL lemmas with their modern counterparts in *Foclóir Gaeilge-Béarla* (Ó Dónaill 1977), as contained in the Modern Irish FST. Implementing lexical-level tag mappings between the Old Irish and Modern Irish FSTs will assist research on the development of verbs and historical roots, making possible a systematic diachronic study of processes such as lexicalisation, relevant for the process of verb stem formation throughout the history of Irish. Possible computational solutions to dealing with grammatical and orthographical variation in Early Irish (i.e., Old and Middle Irish) are also discussed in this chapter.

Acknowledgements

Firstly, I would like to gratefully acknowledge two funding bodies for awarding me with scholarships that enabled me to successfully complete my doctoral degree: Trinity College Dublin (Ussher Fellowship, 2014-2017) and the Irish Research Council (Government of Ireland Postgraduate Scholarship GOIPG/2017/1808).

Thanks also to former and current postgraduate friends and colleagues at South Leinster Street, the Trinity Long Room Hub, and College in general, for creating a friendly and stimulating study environment. I received great support in particular from Frances Brady, Sioḃán O'Brien Green and Tim van Wanrooij. Thank you all.

Dr Eoin Mac Cárthaigh (Department of Irish and Celtic Languages, Trinity College Dublin) provided helpful feedback on an earlier draft in the context of my Ph.D. confirmation process. I am also grateful to Professor David Stifter (Department of Early Irish, Maynooth University) for showing an interest in my project and for making time to discuss matters relating to Old Irish verb morphology with me. Many of my colleagues kindly offered to proofread parts of this thesis, namely, Dr Ciaran McDonough, Dr Katherine Ravenna Morales, Dr Daniel Watson, Dr Ruud Koolen, Alejandra Núñez Asomoza, Antoin Eoin Rodgers and Adrian Doyle. All remaining errors and omissions are of course my own.

My parents have always endorsed my educational choices and career path; I would like to take this opportunity to thank them from the bottom of my heart for their support and advice along the way.

This work is dedicated to two former Dutch mentors who sparked my research interest in language variation and historical linguistics: Dr Ben Hermans, phonologist and dialectologist formerly of Tilburg University and the Meertens Institute in Amsterdam, and Professor Peter Schrijver, chair of Celtic Languages and Culture at Utrecht University. *Heel erg bedankt voor uw inspirerende colleges en begeleiding.*

Last but not least, I would like to express my sincere gratitude to my supervisors Dr Elaine Uí Dhonnchadha and Professor Ruairí Ó hUiginn for providing guidance and generously sharing their knowledge in the course of my project. *Míle buíochas ó chroí*!

Contents

List of Tables **xiii**

List of Figures **xv**

List of Code Examples **xvii**

Glossary **xix**

1 Introduction: background and goals **1**

1.1 Introduction 1

1.2 Historical overview of the Irish language 3

1.2.1 Language stages 3

1.2.2 Orthography 4

1.3 ‘Standard’ Old Irish 5

1.4 Relevant (mainly) printed works for historical Irish 7

1.5 A ‘lexicographical gap’ 9

1.6 Research goals 12

1.6.1 Aims and objectives 12

1.6.2 Scope 14

1.7 Synthesis 14

2 Old Irish verbs: the morphology-phonology interface **17**

2.1 Introduction 17

2.2 The Early Irish verbal complex 17

2.2.1 Basic structure and terminology 17

2.2.2 Vowels and consonants in unstressed position 20

2.2.3 Syncope 21

2.2.4 The augment 22

2.2.5 Pronominal affixes and emphasising particles 23

2.2.6 Relativity 25

2.2.7 Stem formation and endings 26

2.3 Stem allomorphy 28

2.4 Quantifying inflectional variation 30
2.4.1 Monolithic stems 30
2.4.2 Average size of the verbal paradigm 31
2.4.3 Available lists of verbs and verb roots 32
2.4.4 Focus in the present work 34
2.5 The verb in Middle Irish, and beyond 34
2.6 Synthesis 36

3 Computational approaches and methodology 39
3.1 Introduction 39
3.2 Background to computational linguistics 40
3.3 Linguistic tasks and the NLP pipeline 41
3.3.1 Linguistic annotation 41
3.3.2 Pre-processing and orthography 41
3.3.3 Morphology 42
3.3.3.1 Morphological parsing, stemming and lemmatisation 42
3.3.3.2 Finite-State Transducers (FSTs) 43
3.3.4 Part-Of-Speech Tagging 43
3.3.5 Parsing on the sentence level 44
3.4 Rule-based vs. machine-learning methods 44
3.5 Testing 45
3.6 Computational approaches to historical texts 45
3.6.1 Orthographical standardisation 45
3.6.2 Morphological analysis and lemmatisation 47
3.6.3 Part-Of-Speech Tagging (mainly statistical) 49
3.7 Methodology employed in the present work 51
3.8 Synthesis 53

4 Implementation 55
4.1 Introduction 55
4.2 Two-level finite-state machines: transducers 55
4.3 Inventory of the FST toolkit 59
4.3.1 Instruments: finite-state tools used 59
4.3.2 The `lexc` format: building lexicons 59
4.3.3 Regular expression operators 60
4.3.4 Replace rules, composition and alphabet 63
4.3.5 Morphotactic restrictions: filters and flags 64
4.3.6 Regression testing 66
4.4 Modelling the verbal complex: choices and challenges 67
4.4.1 Lexical and surface level description 67
4.4.2 Typographical variability and morphotactic dependencies 67

4.4.3 A solution: two main lexicons 70
4.5 The proclitic lexicon (`proclitic.lexc`) 71
4.6 The stem-and-ending lexicon (`se.lexc`) 73
4.6.1 Monolithic stems: recapitulation 73
4.6.2 Encoding the monolithic stem 74
4.6.3 Stem consonants, endings and suffixes 75
4.6.4 Syncope 82
4.6.4.1 Isolating the right vowels 82
4.6.4.2 Phonotactic restrictions on syncope 84
4.6.4.3 Syncope and consonant quality assimilation 84
4.6.4.4 Analogy 85
4.6.5 A worked-out example: the *f*-future 86
4.6.6 Stem entries for strong verbs 88
4.7 Verb stem dependencies and endings 89
4.7.1 `se.lexc`: absolute vs. conjunct endings 89
4.7.2 Flag diacritics, lexical verb type and dependency 91
4.7.3 Rationale for stem entries on a per-verb/lemma basis 93
4.8 Combining proclitic and stem-and-ending lexicons 95
4.8.1 Concatenating the lexicons 95
4.8.2 Matching mutation tags and underspecified orthography 96
4.8.3 The conjunct particle *no*, passives, and infixed pronouns 99
4.9 Emphasising particles and agreement 100
4.10 Final replacements and word-initial capitals 102
4.11 Synthesis 104

5 Case study: *Táin Bó Fraích* **105**
5.1 Introduction 105
5.2 *Táin Bó Fraích* 105
5.2.1 Purpose of the case study 105
5.2.2 Background to the text 106
5.2.3 Verb count 107
5.3 Small modifications to the transducer 108
5.3.1 Preverbs and infixed pronouns 108
5.3.2 Delenition of θ 108
5.3.3 Updated stems and an old inflectional ending 109
5.4 Pre-processing and `flookup` 109
5.5 Results 111
5.5.1 W1 and W2a verbs 111
5.5.2 Results across all words, and compared to other texts 113
5.6 Discussion 114

5.6.1 Ambiguity, segmentation, editorial policy 114
5.6.2 Issues not covered in the FST modelling framework 116
5.6.3 Middle Irish forms 119
5.6.4 Normalised Old Irish and spelling variation 119
5.7 Lemmatisation 120
5.8 Synthesis 121

6 Conclusions and future work 123
6.1 Introduction 123
6.2 Past and present: bridging the gap 123
6.3 A morphological parser for Old Irish verbs 124
6.4 The future: a roadmap for mapping verbal cognates 126
6.4.1 Two mapping methods 126
6.4.2 Lexical-level mappings using two Finite-State Transducers 126
6.4.3 Lemmatisation and eDIL 131
6.4.4 Normalisation 132
6.4.5 FST adaptation and sequential transducers 132
6.5 Concluding remarks 134
6.6 Synthesis 135

Bibliography 137

Appendix A Survey of digital linguistic resources for historical Irish 147
A.1 Available 147
A.1.1 Lexicons 147
A.1.1.1 electronic Dictionary of the Irish language 147
A.1.1.2 Glosses databases 148
A.1.1.2.1 Milan Glosses database 148
A.1.1.2.2 Priscian Glosses database 149
A.1.1.3 *In Dúil Bélrai* 149
A.1.1.4 *Bunadas* 150
A.1.1.5 *Foclóir Gaedhilge agus Béarla* (Dinneen) 151
A.1.1.5.1 PDF 151
A.1.1.5.2 Irish-English Dictionary online 151
A.1.1.6 *Foclóir Stairiúil na Gaeilge* 151
A.1.2 Corpora 152
A.1.2.1 Corpus of Electronic Texts (CELT) 152
A.1.2.2 *Thesaurus Linguae Hibernicae* (TLH) 153
A.1.2.3 Bardic Poetry corpus 153
A.1.2.4 *Corpas Stairiúil na Gaeilge* 154
A.2 Unavailable 155

A.2.1 Miscellaneous 155
A.2.1.1 Electronic Lexicon of Medieval Irish 155
A.2.1.2 Linking Dictionaries and Texts 157
A.2.1.3 Digital Dinneen 159

Appendix B Old Irish verb paradigms 161

Appendix C Code 175
C.1 Lexicons (`.lexc`) 175
C.1.1 `copula.lexc` 175
C.1.2 `proclitic.lexc` 179
C.1.3 `se.lexc` 184
C.1.4 `tbf.lexc` 194
C.2 Rules (`.rule`) 197
C.2.1 `1_se_filters.rule` 197
C.2.2 `2_se_phon_non_pres_stem_form.rule` 198
C.2.3 `3_se_phon_lowering.rule` 198
C.2.4 `4_se_phon_pal.rule` 199
C.2.5 `5_se_phon_del_stem_vow.rule` 199
C.2.6 `6_se_phon_syncope.rule` 200
C.2.7 `7_se_phon_cons_qual_assim.rule` 201
C.2.8 `8_se_phon_stem_vow.rule` 201
C.2.9 `9_se_orth_vow_pal_cons.rule` 202
C.2.10 `10_se_orth_end_vow.rule` 202
C.2.11 `11_se_vow_emph.rule` 202
C.2.12 `12_se_del_pal_markers.rule` 203
C.2.13 `13_se_phon_orth_cons.rule` 203
C.2.14 `procl_se_filter_1_aug.rule` 204
C.2.15 `procl_se_filter_2_no.rule` 204
C.2.16 `procl_se_filter_3_ipv.rule` 205
C.2.17 `procl_se_filter_4_pass.rule` 205
C.2.18 `procl_se_filter_5_sv.rule` 206
C.2.19 `procl_se_filter_6_mut.rule` 207
C.2.20 `procl_se_filter_7_emph.rule` 207
C.2.21 `v_all_cap_filter_imp.rule` 209
C.3 Scripts (`.script`) 209
C.3.1 `added.script` (regression testing) 209
C.3.2 `added_lower.script` (regression testing) 210
C.3.3 `all.script` 210
C.3.4 `alphabet.script` 210
C.3.5 `cap.script` 211

C.3.6 lost.script (regression testing) 212
C.3.7 lost_lower.script (regression testing) 212
C.3.8 mutation.script 212
C.3.9 proclitic.script 213
C.3.10 procl_se.script 215
C.3.11 se_1_bare.script 216
C.3.12 se_2_abs_conj.script 218
C.3.13 se_3_mut.script 219
C.3.14 v_all.script 220
C.4 Stem entry files (.txt) 222
C.4.1 compoundW1.txt 222
C.4.2 compoundW2a.txt 223
C.4.3 simpleW1.txt 223
C.4.4 simpleW2a.txt 224
C.5 Shell script and directory structure 224

List of Tables

1.1 Medieval and modern stages of Irish. 3

2.1 The interaction between verb type and dependency in Early Irish. 19

2.2 Preliminary schemas representing the Early Irish verbal complex. 20

2.3 Spelling of unstressed vowels in closed syllables illustrated with forms of *beirid* 'carries'. 21

2.4 Schemas representing the Early Irish verbal complex, with addition of the augment. 23

2.5 Schemas representing the Early Irish verbal complex, with pronominal affixes and emphasising particles added. 24

2.6 Schemas of the Early Irish verbal complex, now also showing relativity. 26

2.7 Stem formation for weak and strong verbs. Taken from Stifter (2009: 96). . . . 27

2.8 Stem allomorphy illustrated with prt. 3sg. forms of *léicid* and *do·léici*. 29

2.9 Monolithic stems marked in the Early Irish verbal complex schemas. 30

2.10 Root count based on Schumacher's (2004) *Die keltischen Primärverben*. 33

2.11 Verb roots/lemmas between Old and Modern Irish. 36

4.1 The main regular expression operators used (the preceding command `regex` is not shown). 61

4.2 List of parses including ungrammatical ones (with a strike-through) for the ambiguous orthographical form *níléicisem* (*ní-léic-i=sem*). 69

4.3 Contrasting some third-person singular endings in Stifter (2009) and the encoding in the current implementation. 80

4.4 Cascade of surface-level rules and intermediate strings for deuterotonic fut. 1pl. conj. *·ellfam*/*·eillfem* (*ad·ella* 'approaches, visits'). 87

4.5 Cascade of surface-level rules and intermediate strings for prototonic fut. 1pl. conj. *·aidlebam*/*·aidlibem* (*ad·ella* 'approaches, visits'). 88

5.1 Text statistics for *Táin Bó Fraích*. 107

5.2 Test results for weak verbs including lemmatisation. * = Middle Irish ending. . 111

5.3 Results relative to unique inflected verb forms (excl. verbal nouns) contained in *Táin Bó Fraích*. 112

5.4 Results across all words in *Táin Bó Fraích*. 114
5.5 Morphological analysis results across all words in *Fingal Rónáin* (Greene 1955). 114
5.6 Morphological analysis results across all words in *Orgain Denna Ríg* (Greene 1955). 114
5.7 Morphological analysis results across all words in *Esnada tige Buchet* (Greene 1955). 114
5.8 Morphological analysis results across all words in *Orgguin trí mac Diarmata meic Cerbaill* (Greene 1955). 114

6.1 Lexical-level mappings: a selection of verb forms in *Táin Bó Fraích* and their modern cognates. 128

List of Figures

1.1 Digital resources and their coverage of historical Irish language stages (lighter shades denote lesser digital support). 9

1.2 Schematic overview of linking dictionaries and text (Nyhan 2006a: 257). . . . 12

3.1 Minimum edit distance with Old and Middle Irish cognates. 42

3.2 A Finite-State Transducer (FST) accepting, at final state 8, a set of two-level symbol mappings. 43

3.3 A 'two-pronged attack' to map cognate verb forms in Irish. 52

4.1 The regular expression `a+` represented as a finite-state automaton. 56

4.2 The regular relation `<a:b>` visualised as a finite-state network. 57

4.3 Combining a cascade of alternation rules with intermediary forms into one Finite-State Transducer. Taken from Beesley & Karttunen (2003: 35). 58

4.4 A single all-inclusive or lexical transducer. Taken from Beesley & Karttunen (2003: 36). 58

4.5 The FST implementation of the Old Irish verbal complex: two main lexicons. . 71

4.6 A schematic illustration of intermediate levels during rule application, using an example verb form. 78

4.7 An FST network representation of intermediate levels during rule application, using an example verb form. 79

6.1 Diagram showing diachronic mappings and linking of historical Irish cognate verb forms. 127

A.1 Web interface linking CELT (cf. section A.1.2.1) and the Lexicon of Medieval Irish on the word level with a lookup mechanism (Nyhan 2006a: 255). This tool is not available. 158

B.1 Paradigm for *beirid* (S1a). 162

B.2 Paradigm for *ro·ucca(i)* (W1/W2b), suppletive stem of *beirid*. 163

B.3 Paradigm for *crenaid* (S3). 164

B.4 Paradigm for *crenaid* (S3) (continued). 165

B.5 Paradigm for *do·beir* (S1a). 166

B.6 Paradigm for *do·beir* (S1a) (continued). 167
B.7 Paradigm for *do·ratai* (W2a), suppletive stem of *do·beir*. 168
B.8 Paradigm for *do·uccai* (W2a), suppletive stem of *do·beir*. 169
B.9 Paradigm for *gaibid* (S2). 170
B.10 Paradigm for *gaibid* (S2) (continued). 171
B.11 Paradigm for *léicid* (W2a). 172
B.12 Paradigm for *marbaid* (W1). 173
B.13 Paradigm for *marbaid* (W1) (continued). 174

C.1 Directory structure for generating FSTs. 225

List of Code Examples

4.1 Invoking a `lexc` file in `foma`, which is subsequently compiled and put on the stack. 60
4.2 Basic structure of a lexicon in `lexc` format. 60
4.3 Illustrating various ways of compiling a regular expression in `foma`. 63
4.4 Template for a replace rule in `foma` (or `xfst`); compiles into a transducer. . . . 63
4.5 Snippet of `proclitic.lexc` showing the proclitic lexicons containing the preverbs, conjunct particles and augment *ro*. 72
4.6 Snippet of `proclitic.lexc` showing part of the infixed pronoun lexicons. . . 72
4.7 A sequence of replace rules dealing with the infixed pronoun class A 3sg. masc. / neut. 73
4.8 Stem entry for the simplex *marbaid*, part of `simpleW1.txt`. 75
4.9 Stem entries for the simplex *léicid*, part of `simpleW2a.txt`. 75
4.10 Continuation lexicons for W1 and W2a stem vowels as part of `se.lexc`. 75
4.11 Start of continuation classes in `se.lexc`. 76
4.12 The weak present indicative endings as part of `se.lexc`. 78
4.13 Rewriting ə in closed, unstressed syllables (only applicable with endings following stem consonants). 82
4.14 The syncope-rule in the finite-state implementation. Snippet of `6_se_phon_syncope.rule`. 83
4.15 A replace rule to create alternants with the vowel marked for syncope remaining. Snippet of `6_se_phon_syncope.rule`. 84
4.16 Phonotactic restrictions on syncope. Snippet of `6_se_phon_syncope.rule`. . 84
4.17 Palatalisation rule based on adjoining vowel. Snippet of `4_se_phon_pal.rule`. 85
4.18 Post-syncope consonant quality assimilation. Snippet of `7_se_phon_cons_qual_assim.rule`. 85
4.19 Surface-level rules for *f*-future stem formation. Snippet of `2_se_phon_non_pres_stem_form.rule`. 86
4.20 Monolithic stem entries for the S2 verb *gaibid*. 89
4.21 Monolithic stem entries for the simplex *léicid*, part of `simpleW2a.txt`. 90
4.22 Monolithic stem entries for the compound verb *do·léici*, part of `compoundW2a.txt`. 90

4.23 Rules pertaining to the lexical level to filter out illegal concatenation of absolute and conjunct endings. Snippet of `1_se_filters.rule`. 90
4.24 Illustrating flag diacritics with a random selection of verb forms. 92
4.25 Part of a hypothetical stem-entry list for *ber* if classifying stem entries by root. . 93
4.26 Part of a hypothetical stem-entry list for *ber* if classifying stem entries by root, adding flag diacritics. 94
4.27 Monolithic stem entries for a hypothetical stem-entry list for verbs with root *lēc*. 94
4.28 Concatenating (derivatives of) `proclitic.lexc` and `se.lexc`, defined as variables. 96
4.29 A twofold operation to mark lenition on initial consonants using two variables (transducers). Snippet of `mutation.script`. 97
4.30 A prefixed transducer to deal with digraphs for non-lenited anlaut consonants. Snippet of `mutation.script`. 97
4.31 Unioning the lexicon with conjunct endings (`LEXseConj`) with one subjected to mutations and one with digraphs for non-lenited *anlaut* consonants. 98
4.32 Examples of incompatible 'mutation tags' after concatenating (a derivative of) `proclitic.lexc` and the 'mutated', conjunct forms of (a derivative of) `se.lexc`. 98
4.33 Filtering out incompatible lexical-level nasalisation tags. Snippet of `procl_se_filter_6_mut.rule`. 99
4.34 The rule `procl_se_filter_2_no.rule` filters out incompatible lexical-level combinations with *no*. 99
4.35 The rule `procl_se_filter_4_pass.rule` filters out incompatible lexical-level combinations with passive conjunct endings. 100
4.36 A snippet of `se.lexc` focusing in on the pronominal suffixes and emphasising particles. 101
4.37 Some inflected forms illustrating the restrictions encoded in relation to capitalisation of word-initial letters. 103
5.1 Redefined rule part of `proclitic.script` for vowel coalescence with infixed pronoun 3sg. masc./neut. class A. 108
5.2 Missing delenition rule now part of `13_se_phon_orth_cons.rule`. 109
5.3 Updated (monolithic) stem entries for the compound verb *do·imchella* as part of `compoundW1.txt`. 109
5.4 `unix` commands for creating a wordlist from a text. 110
5.5 Applying a wordlist to a lexical transducer and extract non-recognised strings. . 110
A.1 Snippet of eDIL's XML code for entry *téit* 'goes'. 156
A.2 Snippet of XML code for the restructured DIL entry *téit* 'goes' (Nyhan 2006a: 195–239). 157

Glossary

Note: descriptions and page references are only given for specialist terms that might not be immediately obvious to a non-Old Irish expert.

Abbreviation / tag	**Grammatical term**	**Description**	**Page**
1(P)	first person		
2(P)	second person		
3(P)	third person		
ABS	absolute ending	inflectional ending with independent simple verbs, e.g., prs. 3sg. -*(a)id* in _marb**aid**_	18
AUG	augment	particle communicating perfectivity or potentiality, most commonly *ro*	22
CONJ	conjunct ending	inflectional ending with imperatives, dependent simple verbs, and compound verbs, e.g., prs. ind. 3sg. -*i* in _do·léic**i**_	18
CONJ_PART	conjunct particle	sentence-modifying element prefixed to a verb, e.g., negative *ní* ‘not’, followed by a dependent verb form	18
DEUT	deuterotonic	stem variant of an independent compound verb, with the stress falling on the second element (verb root (VROOT), preverb (PV) or augment (AUG)), e.g., _do·**léic**i_ (here stress on VROOT)	19

EMPH	emphasising particle	enclitic, alternatively called *nota augens*, which enforces (with verbs) the inflectional ending or pronominal infix, e.g., 1pl. -*ni*	23
F(EM)	feminine		
FUT	future		
IMP	imperative		
IPF	imperfective		
IND	indicative		
M(ASC)	masculine		
N(EUT)	neuter		
NEG	negative		
PART	particle		
PASS	passive		
PL	plural		
PRON	pronoun		
PROTOT	prototonic (not currently employed as tag)	stem variant of a compound verb typically in dependent position, with the stress falling on the verb's first element, e.g., prs. ind. 3sg. *·teilci* (PV *to-* followed by the verb root (VROOT) *lēc*)	19
PRS	present		
PRT	preterite		
PV	preverb	lexical element accompanying verb root (VROOT) to form a compound verb, e.g., *do·* (underlying *to-*)	18
REL	relative		
SG	singular		
SUBJ	subjunctive		
SUFF	suffix		

VROOT	verb root	the underlying or historical root of a verb. The stem *léic(ī)*, for instance, consists of the root *lēc* with the corresponding tag VROOT; this is encoded as `lēc +VROOT` on the upper level of the transducer	18
W1	Weak 1	weak verb, class 1, *a*-verbs, e.g., *marbaid*	26
W2a	Weak 2a	weak verb, class 2, subtype a, *i*-verbs, e.g., *léicid*	26

Chapter 1

Introduction: background and goals

1.1 Introduction

The present work aims to enhance understanding of the development of the Irish verb by using digital resources and Natural Language Processing methods. The motivation for focusing on the verb, especially in Old Irish (c. 8th–9th centuries A.D.), resonates with McCone's statement in the foreword in his second edition of the *The Early Irish verb* (1997: xviii):

> Concentration upon the verb was dictated by its generally conceded status as the most difficult and interesting area of Old and Middle Irish morphology and few would deny that an understanding of the Old Irish system's workings and development into and through Middle Irish is a prerequisite for being able to deal with the abundance of Old and Middle Irish texts effectively.

The same author, in his chapter 'Key 'Middle Irish' Developments' (c. 10th–12th centuries), states that '[a]ll varieties of Modern Irish are clearly differentiated from Old Irish by a far-reaching overhaul of the verbal system'. At the same time, digital methods to track these changes are lacking, as will be discussed below. The present work aims to make a contribution to digital support for historical Irish facilitating a more systematic study of the diachronic changes in the Irish verbal system. The statement by Borin & Forsberg (2011: 42), who dealt with Old Swedish, rings very true for my project:

> The motivation for undertaking this kind of work is obviously to make our cultural heritage accessible to the public as well as to provide state-of-the-art tools to researchers who wish to utilize this rich text material as primary research data.

The proposed research will contribute to the emerging field of Digital Humanities, a relatively new and highly interdisciplinary research paradigm in which digital and empirically enhanced methods are systematically applied to the humanities, as well as critically reflected upon (cf. for example Drucker 2013). Piotrowski (2012: 6) states that Digital Humanities projects constitute a paradigm shift in the sense that 'quantitative methods are beginning to be regarded as being on par with qualitative research', the research output being 'no longer tied to the restrictions of the printed medium'. Burdick et al. (2012: 8) identify a first wave between the 1980s and early 2000s, common research interests and aims including textual analysis and cataloguing, the study of linguistic features, an emphasis on pedagogical supports and learning environments, and research questions driven by analysing structured data. Important initiatives included the (still ongoing) Perseus Digital Library Project,[1] going back to the mid-1980s, covering the history, literature and culture of the Greco-Roman world; this project is briefly referred to in section 3.6.2 and section 3.6.3.

Digital Humanities has its roots in Humanities Computing, a field which started in 1949 when Father Roberto Busa, in collaboration with Thomas J. Watson of IBM, set out to create an *index verborum* for the works of St Thomas Aquinas and related authors, totalling some 11 million words. Out of the resulting *Index Thomisticus* (cf. also section 3.6.3), one of the first computerised lexicography projects, arose early-day software to create lemmatised concordances (Hockey 2004). For further information cf. Busa (2004), who details in an inspiring fashion both the chronology of his project and his further—now posthumous—ideas and aspirations. This is not the place to discuss in detail the development of Digital Humanities; for a comprehensive overview of Humanities Computing the reader is referred to McCarty (2005). Schreibman, Siemens & Unsworth (2004) provide an overview of the remit of Digital Humanities, including contributions on linguistics, literary studies and lexicography.

The present study deals with computational strategies in relation to modelling language (change), and will facilitate closer cooperation between the Digital Humanities and computational language processing. It is hoped that insights from historical linguistics will inform the computational methods and, conversely, that the computational methods will assist in discerning language change. Two strands can be identified in this thesis:

1. Historical Irish linguistics: documenting morphological changes in the Irish verbal system.

2. Computational linguistics: linking up cognate verbal forms from different Irish language periods using Natural Language Processing (NLP) tools and online resources.

Section 3.2 provides a more focused overview of the history of computational linguistics/NLP. Despite the long history of Humanities Computing, Piotrowski (2012: 6) has observed that there has been 'surprisingly little communication or collaboration' between Humanities Computing and NLP, although the same author (p. 7) observes that NLP and Digital Humanities are now slowly starting to converge, especially in the field of cultural heritage.

The rest of this chapter is structured as follows. In section 1.2.1 I will provide a short overview of the history of the Irish language. This section is followed by important notes on linguistic variation in Old Irish and the concept of a standard, in section 1.3. The hiatus in both printed and digital support for historical Irish is the subject in sections 1.4 and 1.5, respectively.

[1] `http://www.perseus.tufts.edu/hopper/`.

The final two sections are dedicated to my research goals (section 1.6), including the aim and scope of the work, and a synthesis of matters discussed in this chapter (section 1.7).

1.2 Historical overview of the Irish language

1.2.1 Language stages

The historical period of Irish can be divided into the language stages shown in Table 1.1. Middle Irish posterior's limit is usually taken to be c. 1200; the historical Early Irish period therefore roughly spans the early medieval period up until the Anglo-Norman invasion of Ireland (1169). The period from c. 1200 onwards is, in broad linguistic terms, referred to as 'Modern Irish'. Leaving aside stone inscriptions in the so-called *Ogham* (Early Irish *Ogam*) alphabet dated to the 4th or 5th century A.D., writing in Irish goes back to the 6th century A.D., when the Irish started to write down their native language using the Latin script (Greene 1966: 33). Goidelic is the name for the Gaelic language family which comprises the modern languages of Irish, Scots Gaelic and Manx (Manx became extinct in 1974). The common ancestor of these languages is Old Irish (c. 8th–9th centuries).

Table 1.1 – Medieval and modern stages of Irish. The classification of periods constituting Early Irish is taken from Stifter (2009: 55). Archaic Irish is alternatively called Early Old Irish (Russell 2005). For Post-Classical Modern Irish cf. Ó Háinle (2006). An overview of the entire historical period is provided in Greene (1966) and Russell (2006).

	Language stage	Time period
Early Irish	Archaic Irish	c. 7th century A.D.
	Old Irish	c. 8th–9th centuries
	Middle Irish	c. 10th–12th centuries
Modern Irish	Early Modern Irish (incl. Classical Modern Irish)	c. 13th–mid 17th centuries
	Post-Classical Modern Irish	c. mid 17th–mid 19th centuries
	Irish of the Revival period	late 19th century-early 20th century
	Standardised contemporary Modern Irish	1958-present

Old Irish is 'the earliest period of Irish—or of any Celtic language—for which the extant record is sufficiently full and varied to permit a full synchronic description' (Stifter 2009: 59). This description is largely based on the language of a substantial body of interlinear and marginal glosses accompanying Latin texts[2] in manuscripts produced within the Old Irish period—unlike many other Early Irish texts which are only found in later medieval manuscripts. Thurneysen (1946) is almost entirely based on these Old Irish glosses. The language of the glosses (with the addition of a few other texts) is sometimes referred to as Classical Old Irish (e.g., Russell 2005: 407; cf. also section 1.3).

[2] The three most important collections are: Würzburg (Wb.), Milan (Ml.) and St Gall (Sg.), cf. also section 1.3.

In contrast to the relatively stable and normative phases of Old Irish (but cf. section 1.3 below) and Classical Modern Irish, Middle Irish (c. 10th–12th centuries) represents a language in transition, with some forms adhering to Old Irish norms, some anticipating the literary standard of Classical Modern Irish (13th–mid 17th centuries) and some consonant with neither (McCone 1997: 166).

Early Modern Irish (13th–mid 17th centuries) encompasses Classical Modern Irish, i.e., syllabic verse composed by professional poets relying on patrons in a milieu of Gaelic chiefdoms. In contrast to the regulated grammar of this bardic poetry,[3] early modern prose reflects hugely varying registers, ranging from archaic language to registers that are not far removed from 19th-century Irish (Ó hUiginn 2013).

Post-Classical Modern Irish refers to the literature produced between the Flight of the Earls (1607) and the Great Famine (1845-49), when Irish becomes a language mainly of the rural peasantry, due to English political and economic domination (Ó Háinle 2006). Whereas the poetry in the Classical Modern Irish period is highly standardised, this period is characterised by a more regional orientation in writing, and the coming to the fore of the Irish dialects (Williams 1994: 8).

The period between the Great Famine and the creation of the Free State (1922) is known as the Gaelic Revival, which witnessed an increased production of original work, facilitated by institutions such as the Gaelic League (*Conradh na Gaeilge*), established in 1893 (Mahon 2006). After independence, plans were made for a standardisation of the grammar and spelling for Irish, ultimately codified in a 1958 booklet published by the Government's Translation Department.[4]

1.2.2 Orthography

During the Old Irish period Irish scribes started to come to grips with the fundamental problems associated with Irish spelling, which became based on the orthographical conventions associated with the writing of British Latin (Ahlqvist 1994: 43). This orthographical system, based on the pronunciation of Latin by speakers of British Celtic, was in competition with older conventions that are likely to go back to the spelling system associated with inscriptions on Ogam stones in Primitive Irish (Ó Cróinín 2001: 10–11). A feature of this older system is the spelling of intervocalic voiced plosives /b, d, g/ as *b*, *d* and *g*, respectively, rather than *p*, *t* and *c* as in the 'British' system, which, however, eventually won the day (Ó Cróinín 2001: 10).

Both writing systems can be seen in the oldest Irish glosses, dating from around 750 A.D. An example relating to the above-mentioned intervocalic plosives is archaic *agaldem* vs. later Old Irish *ac(c)aldam* 'conversation', also showing a different vowel in the final syllable (*-em* vs. *-am*) (Ó Cróinín 2001: 12). Other 7th-century Archaic Irish spelling conventions related to stressed vowels can be illustrated with *lóg* for later *lúag/lúach* 'value, worth', as pointed out

[3] Laid down in grammatical tracts, cf. especially tract III (Bergin 1946).

[4] With further revisions in 2012 and 2016, cf. `http://www.oireachtas.ie/parliament/about/rannoganaistriuchain/ancaighdeanoifigiuil/`.

by Ó Cróinín (2001: 14). Cf. section 1.3 for a discussion on 'standard' Old Irish.

It is not until the Classical Modern Irish period (c. 13th–mid 17th centuries) that an orthographical system arose with somewhat more transparent mappings between sounds and letters (for an overview of the phonetics and the marking of initial consonant mutations in the Classical standard cf. McManus (1994: 343–360)). A feature of Classical Modern Irish, for example, is the more consistent and less unambiguous encoding of the velarised or non-palatalised as opposed to palatalised consonants. This distinction in consonant quality is codified orthographically by means of vowels surrounding consonants or consonant clusters. Initial consonant mutations, employed to signify grammatical relations in close syntactic groups, are another typical feature of Irish (and of the Insular Celtic languages in general). According to McCone (1997: 243):

> The orthographical representation of these changes to initial consonants was far from fully worked out in Old or Middle as opposed to later Modern Irish, and a number of ambiguities for which Latin spelling offered no obvious solution were tolerated in writing.[5]

As scribes were able to deploy archaic word forms, many spelling conventions originating in Old Irish are retained in Middle Irish texts (Russell 2006: 990). Orthographic conventions in early modern prose texts range from archaic to 'modern', and differ widely across and within texts (Ó hUiginn 2013). The standardisation of the orthography in 1958 has attempted to simplify the spelling by bringing the written language more in line with the spoken language. For example, many consonant (clusters), while not pronounced, were until recently still written, reflecting the pronunciation of centuries earlier. An example is standard *marú* 'killing', pre-standard *marbhadh* and Early Irish *marbad*.

1.3 'Standard' Old Irish[6]

It should be borne in mind that, although Old Irish texts are commonly believed to show little or no trace of synchronic variation (Stifter 2009: 60), Old Irish by no means represents an entirely stable language period. The concept of a uniform Old Irish language sometimes blinded people to the variation in the data (McCone 1985: 93). In this context it is also worth mentioning recent work by Peadar Ó Muircheartaigh, who has made valuable contributions to the study of the sociolinguistics of Old Irish (Ó Muircheartaigh 2015). According to the latter, the fairly

[5] For a complete overview of the so-called base, lenited and nasalised variants of consonants and phoneme-to-letter correspondences in Old Irish cf. Stifter (2006: 377–378)

[6] At the 2018-2019 O'Donnell Lecture at the University of Oxford (10 May 2019), Professor David Stifter expressed views that contradict some of the ideas discussed here. His ERC-funded *Chronologicon Hibernicum* project (`https://www.maynoothuniversity.ie/chronologiconhibernicum`) has found that there is much more linguistic variation within Old Irish than is commonly assumed; some of this synchronic variation may be diatopic or diastratic. Moreover, according to Stifter, traditional statements suggesting the existence of a literary standard show a partial neglect of the sociolinguistic implications of a standard text language spread over a vast area.

uniform nature of Old Irish is a product of a concentration of scribal networks in the north-east of the island of Ireland, which may explain how a standard was maintained over an extended amount of time.

The occasional spelling deviation found in the glosses, often of an innovatory nature and reminiscent of Middle Irish, is explained by McCone (1985) as a 'lapse' from an educated register into a more colloquial one. This most probably means that the spelling was lagging behind pronunciation. On the other hand, the absence of such variation ('lapses') does not indicate a congruity between literary standard and pronunciation, as a scribe well versed in the norms of the educated register might have been more successful in hiding features of the spoken language. Although diachronic variation might be expected purely on the grounds of date of compilation, which for the Würzburg and Milan Glosses (the earliest of the three main collections) is c. 750 A.D. and c. 800 A.D, respectively (Russell 2005: 412),[7] closer adherence to the educated norm might interfere with evidence for possible diachronic layers.

The present work is not directly concerned with the background for deviations from a 'standard', in other words, whether the variation is diatopic, diachronic or stylistic. It is more concerned with the impact of linguistic variation on the computational encoding of Old Irish verb morphology. This activity becomes significantly more straightforward if one can operate with some sort of standard, whether 'real' or normalised.

My focus in the implementation has been on Old Irish rather than Archaic Irish (c. 7th century) or Middle Irish (10th–12th centuries). Old Irish has received more scholarly attention than any other medieval period—partly because of its normative and reasonably stable nature (cf. sections 1.4 and 1.5). It should be noted, however, that, while adherence to a more or less canonical grammar and spelling was safeguarded by employing grammars/handbooks such as Thurneysen (1946), Strachan (1949)[8] and Stifter (2006) during the implementation phase (Chapter 4), Archaic Irish features such as -/θ/ (e.g., *marba(i)th* 'kills') for 'classical' -/ð/ (e.g., *marb(a)id*) have been silently included as a variant spelling as they are by no means uncommon throughout the Early Irish period (Thurneysen 1946: § 122).[9] In section 2.2.2 I will present some examples of reduction processes in unstressed syllables between Archaic and Old Irish.

Building rule-based linguistic resources on the basis of grammars that are compiled from a language corpus not entirely free from variation is a non-trivial exercise; in this thesis I have not found the time and resources to cater for systematic computational encoding of what one may call normalised or canonical (and superficially 'standard') Old Irish linguistic features, as opposed to Archaic Irish/Early Old Irish or late Old Irish/Middle Irish ones, but this is definitely an area for improvement in the next phase of my project. Section 6.4.5 will explore some possibilities to this end.

[7] The third main collection, the St. Gall Glosses on Priscian, are dated to c. 850.

[8] Originally published in 1904/1905, posthumously edited by Osborn Bergin.

[9] This example illustrates the diachronic development whereby consonants on the word boundary next to an unaccented vowel became voiced, cf. Russell (2005: 429).

1.4 Relevant (mainly) printed works for historical Irish

Scholars currently lack a resource that enables them to easily and systematically track the linguistic developments in the Irish verbal system. At the same time, the verbal system is subject to major changes between Early and Modern Irish (McCone 1997: 165–6). Not all language periods (Table 1.1) are served equally well by grammar books, dictionaries, etc. The following is a list of printed reference works that are important for the study of the Early Irish verb. The list is not meant not be exhaustive (for example, there are numerous text editions which contain valuable linguistic information such as wordlists, etc.) but constitutes relevant and consulted works in the context of this thesis:

- Thurneysen (1946), one of the most comprehensive and authoritative grammars for the Old Irish language.
- Stifter (2006). A widely used introduction to Old Irish.
- Stifter (2009). A concise grammatical overview of Old Irish, accessible to a more general linguistics audience.
- Dictionary of the Irish Language (Quin 1983), and its digitised, partly re-edited and improved online version eDIL (cf. below and section A.1.1.1).
- McCone et al. (1994) (in Irish), and especially the essay on Old Irish and its prehistory by McCone (1994).
- McCone (1997). An extensive synchronic treatment of the Early Irish verb.
- Strachan (1949). A standard reference work on inflectional paradigms, including example inflections for all categories of Old Irish verbs (including full paradigms for a selection of frequent irregular verbs).
- Schumacher (2004). A reference work listing prehistoric roots for a large amount of verbs attested in Early Irish (as well in other ancient and medieval Celtic languages).
- Various lexical databases of the Old Irish glosses, its contents being manually annotated (cf. section A.1.1.2).

On the modern side of things, the following are some important works. This list is also not meant to be exhaustive, especially since the focus of this work on Old Irish and not on (historical) Modern Irish (cf. section 1.6).

- Dinneen (1927). A dictionary for Post-Classical Irish in Gaelic script (section A.1.1.5), pre-dating the official 1958 codification of Irish orthography and grammar (cf. section 1.2.1).
- *Corpas Stairiúil na Gaeilge*, a morphosyntactically annotated digital corpus for the period 1600–1926 (cf. section A.1.2.4).

- Hughes (2008). A reference work for verb paradigms containing a significant amount of contemporary Modern Irish verbs. It does not only list the standard inflections but also the forms for each of the three major dialects, with many non-standard forms pointing to pre-standard inflections.

- Ó Dónaill (1977). A standard dictionary for the contemporary standardised language.[10]

- New English-Irish dictionary (*Foras na Gaeilge*).[11]

Exhaustive works for the language periods 'in the middle' are lacking, as section 1.5 will point out. Admittedly, McCone et al. (1994) constitutes a thorough linguistic overview for each language period. However, the work is a collection of individual contributions and—undoubtedly partly because of this reason—does not provide a comprehensive diachronic analysis of any part of speech. It is therefore not suitable for scholars who are interested in systematically tracking the development of (say) a set of Irish verbs and would like to find all the variant forms and inflections within a certain time frame. Moreover, the work is in Irish, and therefore inaccessible to the wider research community. Furthermore, McCone (1997: 168) points out that '[a]lthough valuable catalogues of the verbal systems of individual Middle Irish texts have been produced, a satisfactory synchronic grammar of Middle Irish in English has yet to be written'.

The online resource eDIL,[12] the electronic edition of the Dictionary of the Irish Language (Quin 1983), the most authoritative dictionary for medieval Irish, has been 'a great boon' (Stifter 2009: 59). Nonetheless, and especially in terms of the objectives formulated in section 1.6, eDIL only takes us so far. The original DIL, on which eDIL is largely based, started in the mid-19th century and is therefore a product of 120 years of work, reflecting various editorial policies and inconsistencies. (e)DIL headwords are not consistently given in their Old Irish form and the lists of inflectional forms under each lemma are far from exhaustive. Despite the enormous benefits for scholars, who now have instant access to a constantly improved and augmented digitised dictionary, eDIL was not deemed suitable as starting point for the goal in this work: creating a morphological parser that incorporates complete paradigms adhering to a clear-cut language period and standard. For an overview of DIL and some limitations of this resource cf. Nyhan (2006a: Chapter 2) and sections A.1.1.1 and A.2.1.1 in Appendix A. I will briefly return to eDIL in sections 1.5 and 1.6.

My project must be seen in the wider context of contributions to the development of language resources and technologies for the under-resourced historical stages of Irish. The current hiatus in grammatical and lexicographical coverage of language periods between Old and Modern Irish has only been very partly resolved with the recent advent of digital resources. In the next section I will explore this hiatus further, focusing on electronic lexicons and corpora.

[10] An electronic version is available at `https://www.teanglann.ie/en/fgb/`.

[11] `https://www.focloir.ie/ga/`.

[12] `http://dil.ie/`.

1.5 A ‘lexicographical gap’

This section serves to illustrate the under-resourced status of historical Irish from a lexicographical perspective. Appendix A provides a detailed description of the digital resources and projects relevant to this work. The resources are plotted on a time scale in Figure 1.1. The picture that emerges is one of fragmentation and lesser digital support for the periods between Old Irish and contemporary stages of Modern Irish. One is thus faced with a ‘lexicographical gap’ in the middle, roughly corresponding to the Early Modern Irish period (13th–mid 17th centuries).

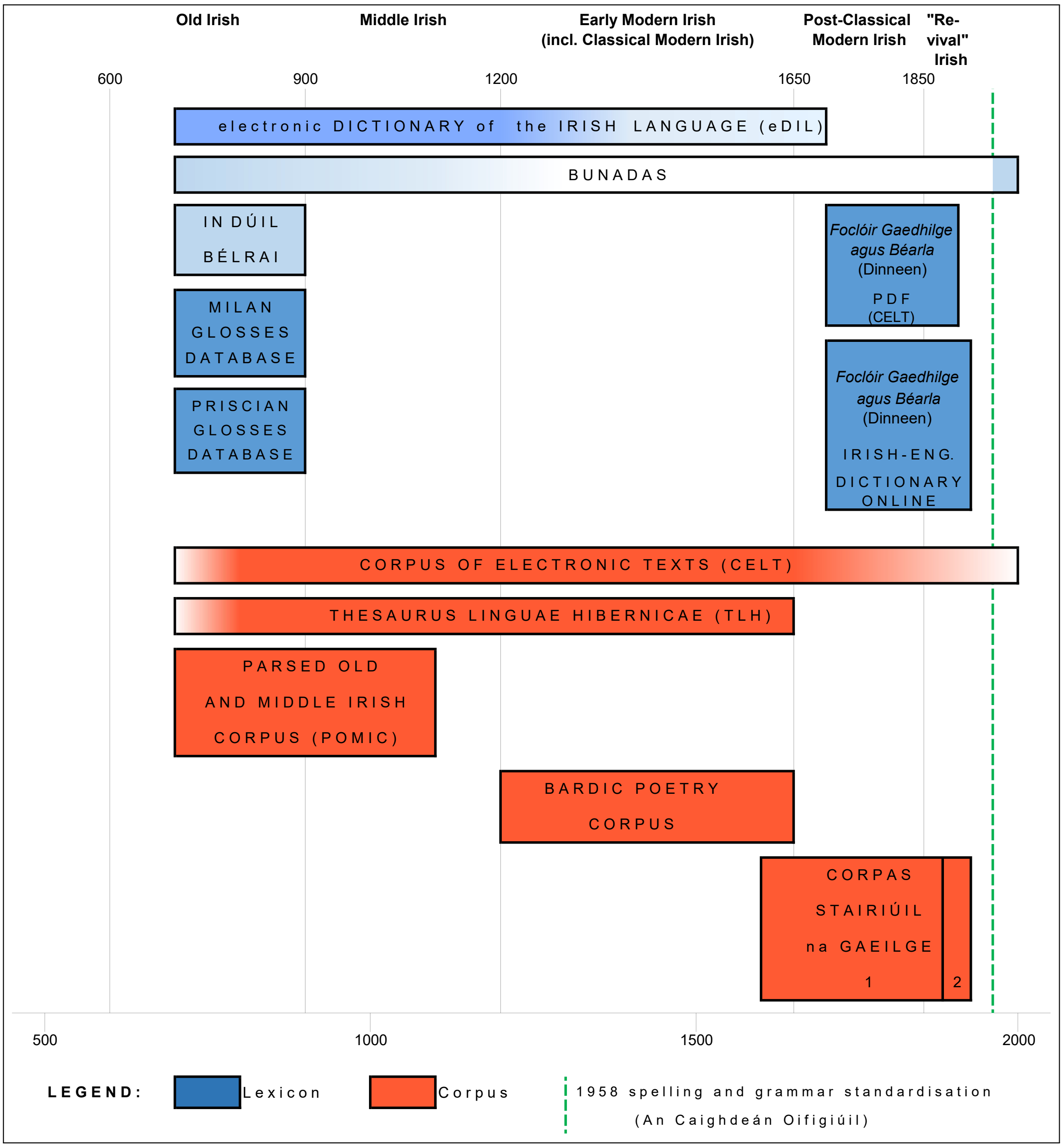


Figure 1.1 – Digital resources and their coverage of historical Irish language stages (lighter shades denote lesser digital support).

Corpora of the Old Irish glosses, the most important contemporary sources for this language period, have been digitised into database format accompanied by detailed morphological glosses (parsing all manually) (Bauer 2014, Griffith & Stifter 2007-2013). In the context of the ongoing ERC-funded *Chronologicon Hibernicum* project at Maynooth University, the aim of which is 'to refine the methodology for dating Early Medieval Irish language developments (c. 6th–mid 10th century A.D.)',[13] various additional collections of glosses are being grammatically analysed and converted into a database format. The material in these databases will be subject to computational methods and statistical analysis in a later phase of the project. *In Dúil Bélrai* 'The Glossary' provides an online English-Old Irish glossary and a database of 5,000 manually lemmatised Old Irish conjugated verb forms.[14]

The Corpus of Electronic Texts (CELT)[15] and *Thesaurus Linguae Hibernicae*[16] (TLH) are the main resources for digitised and machine-readable medieval Irish texts. Neither of these, however, contain linguistic annotation. As far as I know, the only publicly available linguistically annotated corpus for historical Irish is the Parsed Old and Middle Irish Corpus (POMIC) (Lash 2014b).[17] This resource consists of 14 manually tagged and syntactically parsed Old and Middle Irish texts dating from c. 700–c. 1100, using a modified version of the tagging scheme created for the PENN-group of corpora for historical English.[18]

The most important lexicographical resource for historical Irish is the XML-encoded electronic Dictionary of the Irish Language (eDIL), already mentioned in section 1.4, covering the period c. 700–c. 1700. However, the focus of the original hard-copy was on Early Irish (c. 7th–12th centuries)[19] and the headwords are given in their Early Irish, or sometimes Early Modern Irish, shape. This means that one cannot search the dictionary using a modern lemma. The hard-copy edition on which eDIL is based was published in various fascicles for individual letters between 1913-1976, exhibiting different editorial practices and therefore containing many inconsistencies such as variation in the spelling of headwords.[20] However, a revised edition of the electronic version was completed in 2013, remediating some of these issues and incorporating output of recent scholarship. One of the limitations transferred from the original hard-copy is that eDIL is far from exhaustive in listing inflected (verb) forms, as already mentioned in section 1.4 above. It should be added, however, that the original objective of the eDIL project was not to revise the original hard-copy dictionary, but to open up the wealth of information contained in it and to make it accessible to a variety of users (Fomin & Toner 2005).

The most advanced computational techniques in the Irish context are employed in the con-

[13] https://www.maynoothuniversity.ie/chronologiconhibernicum.

[14] http://www.smo.uhi.ac.uk/sengoidelc/duil-belrai/.

[15] https://www.ucc.ie/celt/.

[16] http://www.ucd.ie/tlh/.

[17] Available at https://www.dias.ie/celt/celt-publications-2/celt-the-parsed-old-and-middle-irish-corpus-pomic/.

[18] Cf. https://www.ling.upenn.edu/hist-corpora/.

[19] At the time of compilation of the dictionary, however, Middle Irish was understood by many to extend beyond 1200 A.D. (Breatnach 1994: 221)

[20] An overview of the history of DIL and its limitations is provided in Nyhan (2006a: Chapter 2).

text of the *Foclóir Stairiúil na Gaeilge* 'the Historical Dictionary of Irish' project, which aims at compiling a historical dictionary for the period 1600–2000 on the basis of *Corpas Stairiúil na Gaeilge* 'Historical Corpus of Irish'. This corpus material is being drawn from both oral and written sources and, when complete, is estimated to contain 90+ million words (Uí Dhonnchadha et al. 2014: 13). Currently, one can find all the variants of a modern lemma in segment 1 and 2 of the corpus (1600–1926) through an online query system.[21] While *Foclóir Stairiúil na Gaeilge* is still at the planning stage, there is an intention to use eDIL's page URLs, to link up both resources.[22] One thus currently faces a 'lexicographical gap', meaning that a resource is lacking that facilitates systematic diachronic study of Irish vocabulary and grammar between Early and Modern Irish. This observation has very much prompted the more practical objectives of my thesis, which will be discussed in section 1.6.1.

Although Old Irish is well resourced in terms of both grammars (Thurneysen 1946) and dictionaries (e.g., eDIL, Bauer 2014, Griffith & Stifter 2007-2013), virtually no automatic linguistic analysis tools are available for this period. The fact that no efforts have been made to date in creating a rule-based system to automate morphological parsing is undoubtedly due to the highly complex nature of Old Irish morphology, particularly the verbal system (cf. Chapter 2). At the same time it is my conviction that computational assistance is the only viable way of addressing the lack of linguistic resources, currently inhibiting a systematic study of the development of the verb in Irish.

There have been recent attempts to resolve the discontinuity in historical lexicography and limitations of hard-copy dictionaries. These efforts were all based on XML encoding of preexisting lexicons in combination with web technologies for interlinking. Nyhan (2006a) used XML to encode and restructure a subset of DIL with the aim of retrieving medieval word forms with a high degree of precision. The accompanying unpublished resource is called Electronic Lexicon of Medieval Irish.[23]

Nyhan's resource was envisaged to be part of a larger infrastructure interlinking dictionaries and text (cf. Figure 1.2), and this idea was further explored in Digital Dinneen,[24] an XML-encoded version of Dinneen (1927), which 'will allow a user to follow a modern Irish form in Dinneen's dictionary back its earlier forms in eDIL and the [Electronic] Lexicon [of Medieval Irish]' (Nyhan 2008: 6). An important work for these envisaged links is de Bhaldraithe (1981), who created an index with lemma mappings between Modern Irish (Ó Dónaill 1977) and the corresponding DIL entries.[25] De Bhaldraithe's index is of relevance for the computational approach in the present work in terms of linking cognate verb forms, cf. Section 6.4.3. Digital Dinneen was never finished but has produced a (not publicly available) XML-encoded version of Dinneen (1927).

[21] `http://corpas.ria.ie/`.

[22] Prof. Greg Toner, pers. comm. 10/07/2019.

[23] Only a sample for the letter B is available: `http://research.ucc.ie/lexicon/sample`.

[24] `https://celt.ucc.ie//digineen.html`.

[25] Recently digitised as `droichead` (`https://github.com/kscanne/droichead`) by Scannell (2018), who has included parts-of-speech.

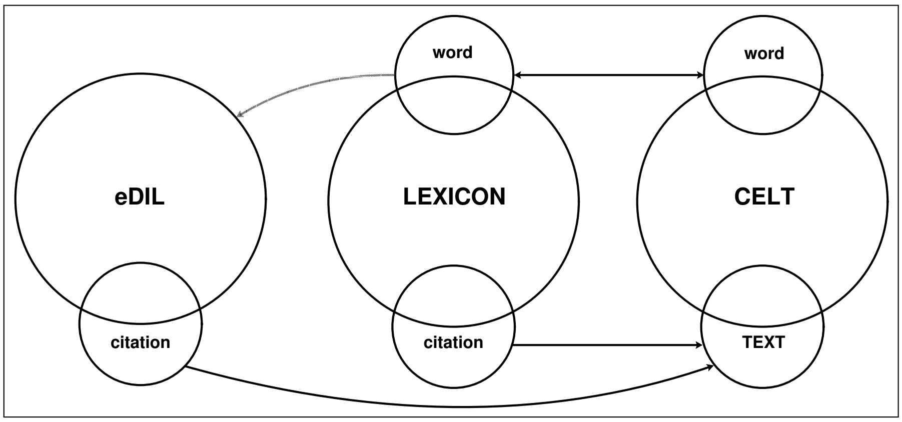


Figure 1.2 – Schematic overview of linking dictionaries and text (Nyhan 2006a: 257).

Dereza (2016) has used a more computationally oriented rule-based approach to automatic lemmatisation for historical Irish. Her Early Irish Lemmatiser uses edit distance algorithms to map unknown or ‘out-of-vocabulary’ Early Irish words in a corpus to their eDIL headword, using extracted lists of inflected forms for each headword in the XML-encoded eDIL. She found that an approximate matching approach is the only way to tackle lemmatisation (Dereza 2016: 13):

> [The] morphophonological complexity compounded by the many non-transparent features of Old Irish orthography makes the traditional dictionary approach to lemmatization with hard-coded lists of possible pseudo-suffixes and rules of their treatment less suitable for Old Irish than for other languages.

This is particularly true for the Old Irish verbal system, the complexity of which will be demonstrated in Chapter 2.

1.6 Research goals

1.6.1 Aims and objectives

The major changes between Early and Modern Irish have been referred to in section 1.1. The exact nature of these changes demands a systematic linguistic investigation; the present work is very much set against the backdrop of these major linguistic developments as well as the hiatus in digital scholarly output on this front, especially lexicographically speaking (section 1.5). The aim of the thesis can be formulated as follows:

- Employ computational methods to facilitate a better understanding of the historical changes in Irish verb morphology.

Establishing the nature of the digital tools already existing has been an important precursor to this aim (cf. Appendix A). As much progress has been made in automatic morphological analysis and POS Tagging for the more recent stages of Modern Irish in the context of *Foclóir Stairiúil na Gaeilge* (1600–2000), and digital support for Early Modern Irish is quite low, it was decided to focus on Old Irish (c. 8th–9th centuries). The primary reasons are the following:

a) This period is relatively well resourced in terms of grammars and (to a somewhat lesser degree) digital resources (mainly eDIL and Old Irish glosses databases, cf. Appendix A).

b) Compared to Middle Irish it shows a relatively stable grammar and orthography and, consequently, it

c) 'furnishes a yardstick with which to assess the abundant literary production of the medieval period' (Stifter 2009: 59).

The initial larger goal of my project was a computational mapping of Old Irish (8th–9th centuries) verbal forms to their Modern Irish cognates. Instrumental for this mapping and hence a major practical aim in the present work, justified above, was the creation of a morphological analyser for Old Irish verbs. However, it became clear in the course of the research that the larger goal was not achievable in the time available due to the complexity of building the morphological parser. Therefore, the subsidiary aim has become the main deliverable of the thesis. What is described in the following is the building of a morphological analyser (and generator) for Old Irish, focusing on verbs. Section 6.4 provides a roadmap indicating what would be needed to map early and modern verb forms.

Although Dereza's (2016) lemmatisation approach is both an extremely valuable contribution to computational methods for Early Irish and part of the framework presented in this thesis (section 3.7), the goal of this thesis is to arrive at a full and detailed morphological parse of a (verb) form, rather than at a dictionary headword in eDIL only; there are two main reasons for this: (1) eDIL is not exhaustive in listing inflected verb forms; (2) a morphological parser, which incorporates all inflected forms, is a fundamental resource for the subsequent automatic linguistic processing activities envisaged as part of this work, and instrumental for systematically and comprehensively linking up Old and Modern Irish cognate verb forms (Section 6.4).

A computational resource that bridges the 'lexicographical gap' between Early and Modern Irish will greatly benefit scholars operating at the intersection of Early and Modern Irish who are now faced with insufficient support to deal with the plethora of intermediate variant forms encountered in manuscripts, ranging from archaic early medieval language use to innovative forms in Early Modern Irish and more contemporary stages of the language. The present work hopes to assist those scholars by providing tools for automatic morphological analysis and lemmatisation, thereby accelerating the work on historical texts. These activities will most certainly be welcomed by scholars of medieval philology, who now have to work with a large amount of Irish-language manuscripts whose texts have not been transcribed or edited, let alone translated. Apart from the envisaged benefits for researchers down at the coalface, i.e., those taking the text from the manuscripts, the present work will also contribute to the creation and advancement of Natural Language Processing methods and digital tools that aid 'pure' linguistic inquiry, including etymological studies, diachronic and synchronic studies on the relation between Old Irish morphology and phonology (cf. Chapter 2), morphosyntactic analysis (by means of Part-Of-Speech Tagging, cf. section 3.6.3), and syntactic parsing.

1.6.2 Scope

Section 2.4 in the next chapter is an important precursor to the implementation. It tries to define the amount of inflectional variation within the verb system by quantifying the average amount of forms per verb, classifying verb types and defining stem variation. There is some literature on verb root and stem classifications, but it turned out that what is lacking is a comprehensive overview of medieval Irish roots/stems, the preverbs that they may take and a stem classification. This is yet another gap in knowledge and research output, and sealing it was not deemed to be feasible in the context of this project. What I aim to do in this thesis, therefore, is to establish an estimate of predictable vs. unpredictable inflection and decide whether the computational paradigm/implementation chosen is worthwhile when set against the balance between manual efforts (including expert knowledge) and automatic methods.

Assessing the balance between automatic and manual methods is an important aspect in the context of a rule-based approach, as is information on scores attainable by automatic morphological analysis. Even when equipped with a rough idea of the balance between predictable verb inflection (a less knowledge-intense implementation) and unpredictable verb inflection (a more knowledge-intense implementation), creating links between all the variants and inflected forms of even a few verb lemmas across the entire historical period of Irish is an enterprise too vast in the context of a Ph.D. project. The amount of verb lemmas in eDIL, for example, is 4,127. It was decided to meaningfully reduce the scope of the research by focusing on Old Irish and, moreover, on a relatively small set of test cases (Chapter 4) within the subset of so-called weak verbs (section 2.2.7 and section 2.4.4); the verbs are *ad·ella* ‘approaches, visits’, *brissid* ‘breaks’, *do·léici* ‘lets go, releases, casts’, *léicid* ‘lets’ and *marbaid* ‘kills’. Paradigms for a selection of verbs, some of which are part of this test set, are provided in Appendix B. The system implemented has been mainly evaluated against 50 inflected forms occurring in the Early Irish text *Táin Bó Fraích*, classified under 27 verb lemmas belonging to the weak classes (the full list is given in Table 5.2 on page 111).

The implementation and evaluation procedure described in this thesis will hopefully pave the way for development of a computational infrastructure with exhaustive lexical coverage: it potentially allows us to extend the framework to all verbs and diachronic variants in a future research project.

1.7 Synthesis

This chapter has provided the background for the thesis. The focus on the Irish verbal system was motivated by the observation that it is highly complex in Early Irish and witnesses huge changes between Old and Modern Irish. At the same time, one can observe a lack of resources that hinders a systematic diachronic study of the Irish verb. Available linguistic resources are surveyed in Appendix A. The hiatus in digital support is most evident with lexicographical resources, which are both fragmented and discontinuous. The primary aim of my thesis is to employ computational resources to better facilitate a historical study of Irish verb morphology, as well as to bridge the 'lexicographical gap'. This work deals with computational morphology for Old Irish since this language period is well resourced and (reasonably) linguistically stable.

An important goal of the project is to assess the accuracy levels attainable through automatic tagging methods applied to historical Irish verb forms. In other words, the goal is to quantify the proportion of computational efforts as opposed to manual work needed to establish satisfactory recognition rates. It is hoped, and indeed expected, that the project will greatly contribute to Irish historical lexicography, philology and linguistics.

Chapter 2

Old Irish verbs: the morphology-phonology interface[*]

2.1 Introduction

As Stifter (2009: 84) has pointed out, the Old Irish verbal system is ‘the most difficult and most challenging section of Old Irish grammar’. In this chapter I aim to outline the most important points, providing a compact introduction to the grammar of the verb for those readers who do not have a background in Early Irish. At the same time it serves to illustrate the impact that phonology has on Old Irish verb morphology and paves the way for the methods used and choices made in the implementation, in Chapter 4. I will first give an overview of the Old Irish verbal system in section 2.2, introducing the notion of *verbal complex*, borrowed from McCone (1994: 1–19). Section 2.3 takes all the introduced concepts together and deals with the nature of stem allomorphy. The goal of section 2.4 is to quantify inflectional variation. In this section I will introduce a computationally workable definition of verb stem. Section 2.5 provides the reader with a general sense of the developments that determined the shape of the verb in Modern Irish. A synthesis follows in section 2.6.

2.2 The Early Irish verbal complex

2.2.1 Basic structure and terminology

As McCone (1997: 17) has pointed out, ‘[l]ike Modern Irish and Scots Gaelic, Old Irish is a basically verb-initial language in which the order verb-subject-object (VSO) predominates, except in the case of clitic pronominal infixes or suffixes’ (cf. section 2.2.5). However, as in Modern Irish, additional structures are found, especially with regard to the subject position in Old Irish (cf., e.g., Mac Coisdealbha 1998, Lash 2014a).

The verbal complex (McCone 1997: 1–19) comprises everything that falls within the ac-

[*] Parts of this chapter are based on Fransen (forthcoming).

centual domain of the verb. The verbal complex has agglutinative features: it may contain, apart from the verb stem and ending, conjunctions, lexical preverbs, particles, and various pronominal affixes. Old Irish does not know independent subject pronouns except with the copula. Most forms therefore are synthetic: person/number is encoded in the verb ending. A single morpheme denotes person and number, and sometimes also tense and mood (for example, prs. ind./subj. abs.[1] 3sg. *-(a)id*). In other words, verb endings are fusional. Third person forms—from the viewpoint of word-based parsing—are inherently ambiguous in that there might or might not be an independent subject. Examples of such forms are glossed without the pronoun in the English translation. The citation form of the verb in Early Irish is prs. ind 3sg. (independent, cf. below).

There is an important distinction between simple and compound verbs. Simple verbs consist of a verb stem and ending only, while a compound additionally takes up to four preceding lexical preverbs, e.g., *ind·árban* (*ind-ad-ro-uss* followed by the verb stem *ben*) (McCone 2005: 72). Preverbs originate in prepositions,[2] modifying the meaning of the verb root, as can be seen in the opposition between (1) and (2), the latter incorporating the lexical preverb *do*. The first of three glossing lines in my examples reflects a diachronic layer which is often needed to explain the surface or orthographical form (cf. section 2.3). These underlying forms are part of the two-level implementation, for which cf. section 4.4.1. A glossary for the abbreviations is found at the beginning of the present work, on page xix.[3]

(1) **beir-id**
ber-SUFF
carry-PRS.IND.3SG
'carries'

(2) **do-beir**
to-ber
PV-carry.PRS.IND.3SG
'brings'

Two different ending sets exist. Absolute endings only occur with simple verbs in absolute (clause-initial) position, while conjunct endings are employed when a preverb (with compounds) and/or an invariably proclitic element (a conjunct particle) precedes the verb.[4] The same conjunct ending set applies for simple and compound verbs. The term dependency is used for verb forms preceded by a conjunct particle. Although dependent verb forms invariably take the conjunct ending set, conjunct endings do not necessarily equal dependency: compound verbs always have conjunct endings, whether independent or dependent. The inter-

[1] Absolute, cf. below.

[2] Indeed, Thurneysen (1946: §§ 819–856) uses 'preposition' for what is now more commonly called (lexical) preverb.

[3] Linguistic annotation in examples in this chapter adhere to the Leipzig conventions for interlinear morpheme-by-morpheme glosses (`https://www.eva.mpg.de/lingua/resources/glossing-rules.php`).

[4] One such particle is *ní* 'not'; for a full list of the conjunct particles cf. Thurneysen (1946: § 38.2).

action between verb type and dependency (on a preceding conjunct particle) translates into the possibilities shown in Table 2.1.

Table 2.1 – The interaction between verb type and dependency in Early Irish, resulting in the employment of two different ending sets and—with compound verbs—two inflectional variants.

Verb type	Independent	Dependent
simple	absolute	conjunct
compound	conjunct (deuterotonic)	conjunct (prototonic)

A rigid stress boundary (phonology), in combination with dependency (morphosyntax), causes divergent inflectional patterns. By default the stress is on the verb root, unless a proclitic element precedes, in which case the stress falls on the second element of the verbal complex. This automatically creates a juncture between an unstressed element (a proclitic) and the stressed or tonic part of the verbal complex. When a preverb occupies the first position in the verbal complex it is realised as a proclitic, which causes the compound verb to be stressed on its second element. This inflectional variant is therefore known as the *deuterotonic* form. The stress is now on the verb root or, in the case of compounds with multiple preverbs, the second preverb.[5]

A conjunct particle is always realised as a proclitic. When it is followed by a now dependent compound verb, the first preverb of this compound comes under the stress as it is the next element in line (hence *prototonic*, stress on the first element). In other words, in dependent compound verbs 'the stress shifts one position to the left onto the first preverb' (Stifter 2009: 89). Compare the deuterotonic form in (2) with the prototonic one in (3). Due to the presence of *ní* in the latter form, the stress shifts from the underlying root *ber* to the first preverb, which is historically/underlyingly *to*, surfacing as *ta-* here.[6] Deuterotonic and prototonic forms not infrequently vary quite considerably.

(3) **ní-ta-bair**
PART-to-ber
CONJ_PART_NEG-PV-carry.PRS.IND.3SG
'does not bring'

It is helpful to introduce the terms *minimal* and *binary* (McCone 1997). Minimal forms are verbs without a lexical preverb or preverbal particle. Generally speaking, this category consists of the simple verbs unaccompanied by a conjunct particle, i.e., with absolute ending inflection. Binary forms include everything else. The minimal-binary classification therefore transcends the distinction between simple and compound: both simple verbs with a conjunct particle and compound verbs are binary, taking the conjunct ending set. Imperative forms always take conjunct endings, and compound verbs assume their prototonic form in the imperative regardless

[5] Ignoring, for the moment, the possibility that the augment (cf. section 2.2.4) occurs in stressed position. Note also that the verb root in a compound verb with more than one preverb is never stressed.

[6] Cf. Appendix B, page 166.

of the presence or absence of a negative imperative particle. However, the deuterotonic form is employed instead in the case of an infixed object pronoun (cf. section 2.2.5), which needs a preceding proclitic element (in this case a pretonic preverb) to attach itself to. Compound verbs with prevocalic *to*, *fo* and *ro* are an exception in that they often assume their prototonic form also in independent (and not necessarily imperative) contexts (McCone 1997: 3).

The above-mentioned discussion and examples have shown that the proclitic juncture is a morpheme boundary that simultaneously acts as a rigid stress boundary, the element immediately after being stressed. As such, it is also important for the workings of syncope: deletion of vowels as a consequence of the stress system, cf. section 2.2.3. The mid-high dot '·' is most commonly used for the proclitic juncture in binary forms. Alternatively, a hyphen, colon (McCone 1997) or whitespace is used. Apart from proclitic prefixes (e.g., conjunct particles and preverbs), Old Irish has proclitic infixed pronouns and post-tonic pronominal and emphasising suffixes, discussed in section 2.2.5. A preliminary overview of the different schemas of the verbal complex for simple and compound verbs is presented in Table 2.2.

Table 2.2 – Preliminary schemas representing the Early Irish verbal complex. E = ending.

Schema	Dependency	Verb type
VROOT $\mathrm{E}_{\mathrm{ABS}}$	independent	simple
CONJ_PART · VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	simple
PV_{1} · ($\mathrm{PV}_{2\text{-}4}$) VROOT $\mathrm{E}_{\mathrm{CONJ}}$	independent	compound
CONJ_PART · PV_{1} ($\mathrm{PV}_{2\text{-}4}$) VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	compound

2.2.2 Vowels and consonants in unstressed position

Other developments mainly in the diachronic sphere are changes to consonants and vowels in unstressed position. These changes can be characterised as phonetic reduction. The preverb *to* (with variant *tu*), for example, is still found as such in unstressed position in Archaic Irish (e.g., *tu·thēgot*[7] 'who come'), but later on invariably *do* (or *du*) (cf. example (4a)[8] and Thurneysen 1946: §§ 178, 855). This creates ambiguity, as the preverb *dī* is also reduced to *do* (or *de*, *du*) in unstressed position, e.g., *do·éc(c)ai* 'looks at' (compare also prs. subj. pass. 3sg. *du·écastar* and prs. ind. pass. 3pl. *de·éctar* of the same verb (Thurneysen 1946: § 831)).

Unstressed vowels in closed syllables (Thurneysen 1946: § 102) are reduced to schwa. Table 2.3 shows how these vowels appear in the orthography (Thurneysen 1946: § 102). The spelling of unstressed vowels is interrelated with syncope (cf. section 2.2.3), which may result in a change of quality of surrounding consonants (becoming either non-palatal or palatal). The spelling of unstressed vowels in closed syllables is 'actively' encoded in the implementation (cf. Code Example 4.13 in section 4.6.3) by using underlying, phonological representations.

[7] Found in the Cambrai Homily (Stokes & Strachan 1901–1910: ii 247.17).

[8] And the other deuterotonic forms of *do·beir*, given in Appendix B, pages 166–169.

Table 2.3 – Spelling of unstressed vowels in closed syllables illustrated with forms of *beirid* 'carries'.

Environment	Spelling of vowel	Example	Morphological gloss
C ə C	*a*	(*ní*)·*ber-**at***	carry-PRS.IND.3PL
C ə C′	*(a) i*	*ber-**(a)id***	carry-PRS.SUBJ.3SG
C′ ə C	*e*	*ber-**ed***	carry-IMP.3SG
C′ ə C′	*i*	*be(i)r-**id***	carry-PRS.IND.3SG

This reduction of vowels to schwa in post-tonic syllables can be illustrated with a development such as ·*berat* (cf. Table 2.3) from prehistoric *-berod* < *-beront* (McCone 1994: 141), Old Irish final *t* representing /d/. Not only interior vowels but also final consonants are liable to variation; for instance, *th* 'alternates frequently with *d* (= ð)' (Thurneysen 1946: § 122). Note also that spelling variation occurs with *berid*; this form may denote either *beirid* (prs. ind. 3sg.) or forms such as *beraid* (prs. subj. 3sg.) (McCone 1994: 80); Old Irish spelling may be ambiguous as to the quality (palatal vs. non-palatal, respectively) of consonants.

2.2.3 Syncope

A key feature of the Old Irish stress system is syncope, the deletion of vowels in even-numbered (but not in final) syllables. In verbal forms, the syncope rule operates counting from the stressed part of the verbal complex, which equals the VROOT with an independent minimal form (typically an independent simplex), and the syllable immediately following the proclitic juncture in a binary form (which may be VROOT, preverb (PV) or augment (AUG); for the latter cf. section 2.2.4).

Syncope may result in changes to consonant quality (i.e., palatalisation or non-palatalisation) depending on the vowel being lost and often results in stem and ending variation. Consider (4a), with the dependent (prototonic) form in (4b), both forms with underlying *to-ber*. The deletion of the root vowel *e* in *ber* in (4b) is due to a higher underlying syllable count. This causes subsequent changes to consonant quality, affecting the ending (for the complete paradigm cf. Appendix B, pages 166–167). In the dependent (prototonic) form in (5b), the preverb also contributes to a larger underlying syllable count compared to (5a) (both *to-lēc*), not only causing syncopation of the root vowel *ē* (<éi>), but also the surfacing of <i>. This vowel is encoded as part of the stem in my approach (cf. section 4.6.2), to generalise across the inflection patterns of the so-called weak verbs (cf. section 2.2.7) focused on in this thesis.

(4) a. **do-ber-am**
to-ber-SUFF
PV-carry-PRS.IND.1PL
'we bring'

b. **ní-tai-br-em**
PART-to-ber-SUFF
CONJ_PART_NEG-PV-carry-PRS.IND.1PL
'we do not bring'

(5) a. **do-léic-set**
to-lēc-SUFF
PV-let-PRT.3PL
'let go'

b. **ní-tei-lc-iset**
PART-to-lēc-SUFF
CONJ_PART_NEG-PV-let-PRT.3PL
'did not let go'

There are numerous exceptions to syncope, many of which are documented in Ó Crualaoich (1999). In fact, it is doubtful whether the example in example (5b) is actually what one expects to find. Ó Crualaoich (1999: 97–98), who discusses irregular syncope, reports on the deletion of the vowel in the third syllable in compounds with root *lēc*, as with verbal noun *teil*†*c*†*thi*[9] and 3pl. aug.[10] prt. *·rel*†*c*†*set*.

2.2.4 The augment

The *augment*, e.g., *ro*, supplies either a resultative or potential meaning, depending on the tense and/or mood of the verb form that it occurs with. Examples (6) and (7) (augmented form of *as·beir*) illustrate this, respectively. The augment is a preverb in origin and for this reason adheres to a positional hierarchy of preverbs, tentatively formulated in McCone (1997: 89–90). The augment is most commonly *ro*, occupying position 4 in this positional hierarchy. The other augments are *ad* (position 3) and *cum*/*con* (position 4), also originally preverbs. However, the latter only co-occur with a limited set of (lexical) preverbs, thus being more restricted.[11] According to McCone (1997: 91), augments have a 'modificatory function that belongs to the grammar of Old Irish and not to its lexicon'. This explains why in the present work the augment is encoded as AUG, and not as PV. The verbal complex schemas incorporating the augment are found in Table 2.4.

(6) **ro-léic**
ro-lēc
AUG-let.PRT.IND.3SG
'has let'

(7) **as-ro-bair**
ess-ro-ber
PV-AUG-carry.PRS.IND.3SG
'can say'

[9] The dagger symbol denotes syncopated vowels.

[10] Augmentation is dealt with in section 2.2.4.

[11] *ad* and *cum*/*con* are underlying forms, with many different surface realisations.

Table 2.4 – Schemas representing the Early Irish verbal complex, with addition of the augment. E = ending.

Schema	Dependency	Verb type
VROOT E$_{\text{ABS}}$	independent	simple
AUG · VROOT E$_{\text{CONJ}}$	independent	augmented simple
CONJ_PART · (**AUG**) VROOT E$_{\text{CONJ}}$	dependent	(augmented) simple
PV$_1$ · PV* (**AUG**) PV* VROOT E$_{\text{CONJ}}$	independent	(augmented) compound
CONJ_PART · PV* (**AUG**) PV* VROOT E$_{\text{CONJ}}$	dependent	(augmented) compound

The position of the augment *ro* is 'a highly complex question' (Stifter 2006: 256) and adds to the already abundant allomorphic variation seen with compound verbs. Moreover, already in the Old Irish period, and during the Middle Irish period, *ro* is gradually adopting the status of conjunct particle, which greatly reduces the amount of allomorphs of this particle (cf. section 2.3, Table 2.8 for its varying shapes). This development runs parallel to other processes of reorganisation and simplification of the verbal system, most importantly the univerbation of compound verbs, i.e., preverbs becoming inseparable from the verb root (cf. also section 2.5). For a detailed discussion of the augment, the reader should refer to McCone (1997: 127–161).

2.2.5 Pronominal affixes and emphasising particles

The skeleton of the verbal complex outlined in Table 2.4 allows for incorporation of affixed pronominal elements, mostly functioning as the object of the verb. With independent simple verbs, that is, minimal forms with absolute endings, a pronominal object may be realised as a suffix, as in (8) (compare with (1)). The underlying morphemes are *ber-ith-us*, with syncope of *i* in the second syllable.

(8) **ber-th-us**
ber-SUFF-SUFF
carry-PRS.IND.3SG-PRON.3SG.F[12]
'carries her'

A more common way is to employ the binary verbal complex, with infixation as a proclitic element, which is positioned after the 'prefix' string, as in (9). Since no 'prefix' is present to facilitate infixation, the semantically empty conjunct particle *no* is employed. Example (8) and (9) are semantically equivalent. The negated version in (10), which is inherently binary due to the presence of the preverbal particle *ní*, can only take the pronoun in the form of an infix and is thus formally similar to (9).

Three different classes of infixed pronouns exist (for an overview and concise discussion cf. Strachan 1949: 26–27). The choice between the first two, A and B, is demanded by phonology; particles and preverbs (originally) ending in a vowel take class A, while those ending in a consonant take class B. Class C is demanded by syntax: when the verb form in question is relative

[12] *or* carry-PRS.IND.3SG-PRON.3PL 'carries them'.

(cf. section 2.2.6) or after the interrogative particle *in*. Infixed pronouns are often accompanied by consonant mutations that transgress the proclitic juncture. The 3sg. fem. (or 3pl.) pronominal infix *-s*, for example, optionally causes *nasalisation* of the root-initial consonant *b* (*mb*), as shown in (9) and (10).

(9) **no-s-(m)beir**
PART-INFIX-ber
CONJ_PART-PRON.3SG.F-carry.PRS.IND.3SG[13]
'carries her'

(10) **ní-s-(m)beir**
PART-INFIX-ber
CONJ_PART_NEG-PRON.3SG.F-carry.PRS.IND.3SG[14]
'does not carry her'

Emphasing particles or *notae augentes* occur in conjunction with (and obligatorily agree with) personal endings and infixed pronouns. These particles are used with both minimal and binary forms, and occupy the very last slot in the verbal complex. The addition of an emphasising particle does not cause syncope (Stifter 2009: 80). An example is provided in (11).[15] Simple verbs cannot have both a suffixed pronoun and an emphasising particle, and absolute relative endings (cf. section 2.2.6) only allow an emphasising particle, not a suffixed pronoun. The verbal complex schemas now incorporating pronominal affixes and emphasising particles are found in Table 2.5.

(11) **no-nn-birt=ni**
PART-INFIX-ber-SUFF=SUFF
CONJ_PART-PRON.1PL-carry.PRT.2SG=EMPH.1PL
'you carried *us*'

Table 2.5 – Schemas representing the Early Irish verbal complex, with pronominal affixes and emphasising particles added. E = ending.

Schema	Dependency	Verb type
VROOT E$_{\text{ABS}}$ (**PRON**) (**EMPH**)	independent	simple
AUG (**PRON**) · VROOT E$_{\text{CONJ}}$ (**EMPH**)	independent	augmented simple
CONJ_PART (**PRON**) · (AUG) VROOT E$_{\text{CONJ}}$ (**EMPH**)	dependent	(augmented) simple
PV$_1$ (**PRON**) · PV* (AUG) PV* VROOT E$_{\text{CONJ}}$ (**EMPH**)	independent	(augmented) compound
CONJ_PART (**PRON**) · PV* (AUG) PV* VROOT E$_{\text{CONJ}}$ (**EMPH**)	dependent	(augmented) compound

[13] *or* CONJ_PART-PRON.3PL-carry.PRS.IND.3SG 'carries them'.

[14] *or* CONJ_PART_NEG-PRON.3PL-carry.PRS.IND.3SG 'does not carry them'.

[15] Although *-t-* may be regarded as the preterite stem consonant, the inflection for this specific person/number ending is *raising*, root vowel *e* becoming *i* (*ber* → *bir*). I have therefore not separated out the preterite stem consonant in the glossing, as this would suggest that this morpheme on its own communicates the 2sg., which is not the case. For the full paradigm of *beirid* cf. Appendix B, page 162.

2.2.6 Relativity

Old Irish does not have inflectable relative pronouns like English ‘which’, ‘whose’ and ‘whom’. Relativity is encoded within the verbal complex alone. Old Irish uses

> a number of completely different strategies . . . , depending on the person, the dependence/independence of the verb, the infixed pronoun, the relation of the subordinate to the superordinate clause, and the syntactical category of the relativized phrase (Stifter 2006: 165).

Special relative endings exist for absolute endings (with simplexes) in the third persons and first person plural, e.g., (12). In plural relative forms, when the rules of syncope demand it, or, in the case of the 3pl. endings, for no apparent reason, a vowel appears before the ending (Stifter 2006: 166). The latter is illustrated with the variant *marbaite* in (13).[16] Doublets are also found with 3pl. passive forms, whether relative or not, e.g., prs. ind./subj. 3pl. pass. *·marb(a)tar*, pass. rel. (abs.) *marb(a)tar* (cf. Appendix B, page 173).[17]

(12) **marb-as**
marb-SUFF
kill-PRS.IND.3SG.REL
‘who kills, that . . . kills’

(13) **marb-aite/marb-tae**
marb-SUFF
kill-PRS.IND.3PL.REL
‘who kill, that . . . kill’

With binary complexes, relativity is marked by a consonant mutation caused by a relative particle which otherwise does not surface, except with the preverbs *ar* and *imm*, appearing as *are* and *imme*/*imma*, respectively (as in (14)). Mutations apply to the following infixed pronoun (if present) or the first stressed syllable after the proclitic juncture. The ‘leniting relative clause’ is used when the antecedent is the subject of the relative clause (as in (14)), or in case of a neuter pronoun functioning as the object (Stifter 2006: 169). A ‘nasalising relative clause’ may be found with a range of temporal, causative and modal adverbial antecedents (Thurneysen 1946: §§ 492–502), and may be employed, as an alternative to the leniting relative clause, when the

[16] Green (1995), whose paradigms are used in Appendix B, only provides non-syncopated *marbaite* (page 173, 3p, below (first) rel in the prs. ind. and subj. paradigm). The initial consonant *t* of the ending is always palatalised in the non-syncopated variant. <t> (/d/) may be written <d> if it immediately follows a consonant, i.e., alternatively ‘syncopated’ *marbdae* (Stifter 2006: 166). The potentially syncopated *a* and *i* of W1 and W2 verbs, respectively (cf. section 2.2.7), here glossed as part of the ending, are computationally encoded as part of the stem, e.g., *marbā*, for which cf. section 4.6.1.

[17] Prs. ind. 3pl. pass. (non-rel.) *do·léicetar* (as opposed to *do·léic*†*ter*) has come up during testing in the context of a case-study using the text *Táin Bó Fraích*, in Chapter 5. This form (spelled *Dolléicetar*) is found under the W2a lemma *do·léici* in Table 5.2 on page 111. Similarly prs. ind./subj. 3pl. pass. (·)*léicter*, (·)*léicetar* (conj. or abs. rel.) of the simplex *léicid* (cf. Appendix B, page 172). The computational implementation of optional syncope is described in section 4.6.4.

antecedent is the object. In the course of Old Irish, the nasalising and leniting type of relative clause is marked also on simple verbs, e.g., the nasalising relative *amal ṅguides* 'as he prays' (otherwise *guides*) (Thurneysen 1946: §§ 495 (b), 504 (c)). Note that, due to the underspecified nature of Old Irish orthography, relativity is often not explicitly marked.

(14) **imm-e-thét**
imbi-INFIX-tēg
PV-REL-go.PRS.IND.3SG
'who goes around'

The negative relative particle is *nád*, *ná*, which changes to *nach* or *nách* for pronominal infixation purposes, although *nád* is also attested in this function (Thurneysen 1946: § 419). Simple verbs without a preceding particle acquire the conjunct particle *no* for relative forms outside the third persons and first person plural. Example (15) shows a relative form of a simple verb with pronominal affixes discussed in section 2.2.5.[18] Note that the conjunct particle *no* appears in the first place to facilitate a relative construction (there is no special absolute relative ending for 1sg.) and, secondly, to support the infixed pronoun, on which relativity is marked.[19] A simple verb with an absolute relative ending does not allow a pronominal suffix. The verbal complex schemas incorporating the relative elements are given in Table 2.6.

(15) **((h)óre) no-ndob-mol-or=sa**
((h)óre) PART-INFIX-mol-SUFF=SUFF
(because) CONJ_PART-REL\PRON.2PL-praise-PRS.IND.1SG=EMPH.1SG
'(because) I praise ye' (non-relative -dob-)

Table 2.6 – Schemas of the Early Irish verbal complex, now also showing relativity. E = ending.

Schema	Dependency	Verb type
VROOT E$_{\text{ABS}\ (\textbf{REL})}$ (PRON) (EMPH)	independent	simple
AUG (**REL**) (PRON) · VROOT E$_{\text{CONJ}}$ (EMPH)	independent	augmented simple
CONJ_PART (**REL**) (PRON) · (AUG) VROOT E$_{\text{CONJ)}}$ (EMPH)	dependent	(augmented) simple
PV$_1$ (**REL**) (PRON) · PV* (AUG) PV* VROOT E$_{\text{CONJ}}$ (EMPH)	independent	(augmented) compound
CONJ_PART (**REL**) (PRON) · PV* (AUG) PV* VROOT E$_{\text{CONJ}}$ (EMPH)	dependent	(augmented) compound

2.2.7 Stem formation and endings

Apart from hiatus verbs, with roots ending in a vowel, Old Irish exhibits an opposition of weak (W1-W2) and strong verbs (S1-S3), which are classified on the basis of their present stem

[18] Found in the Würzburg glosses, 14c18 (Kavanagh 2001: 665, s.v. *molaid(ir)*).

[19] A nasalising relative clause is enforced by the conjunction *(h)óre* 'because' preceding in the text, causing initial *d* of the infixed pronoun to become *nd*. The nasalisation is marked according to the Leipzig glossing conventions for consonant mutations, employing a backslash (REL\); cf. example (17) on `https://www.eva.mpg.de/lingua/resources/glossing-rules.php`.

formation (McCone 1997). The weak and strong classes can be further divided in subclasses (Stifter 2006: 382). The weak verb classes W1 and W2 are also referred to as *a*- and *i*-verbs, respectively (Thurneysen 1946: §§ 521–525, 546),[20] which is important for the implementation: these vowels are computationally encoded as part of the stem; cf. section 4.6.1. The W2 class consists of two members, W2a and W2b. The latter consists of causatives with *-u-* in their present stem.

Old Irish verbs have five stems: present, subjunctive, future, preterite and preterite passive. Stem formation with weak verbs is through suffixation only, and is predictable. Strong verbs show a combination of suffixation, vowel alternations (ablaut) and reduplication. Non-present-stem formation correlates only weakly with the distribution of present stems (Stifter 2009: 96–97). Stem formation types for weak and strong verbs are found in Table 2.7.

Table 2.7 – Stem formation for weak and strong verbs. Taken from Stifter (2009: 96).

	Weak	**Strong**
Subjunctive stem	*a*	*s, a*
Future stem	*f*	*s, e*, reduplication
Preterite stem	*s*	*s, t*, ‘suffixless’: reduplication/long vowel
Preterite passive	= prs. + *-th*	ablaut + dental

Not only is there are weak correlation with strong verbs between present stem type and non-present stem formation types, one needs to know the underlying root to arrive at the stem. For example, the root of the verb *crenaid* ‘buys’ is *crī* (the full paradigm of this verb is found in Appendix B, pages 164–165). To form a preterite stem, a long vowel has to be inserted between *c* and *r*, which is either *íu* or *é*, depending on the ending in the preterite paradigm (Thurneysen 1946: 462).[21] Stifter (2009: 383) identifies this particular root with the template √CRī, R standing for resonant (/l, n, r/). Diachronically, this stem formation is the result of reduplication and subsequent compensatory lengthening: *crī* → *cechr-*/*cichr-* → *cér-*/*cíuir-* (Thurneysen 1946: §§ 71, 691 (a)). The suffixless preterite formation contains a dozen or so different subclasses, depending on the abstract root shape, some containing only a few verbs.

There are six groups of endings which are not arbitrarily combinable with the five stems (cf. Stifter 2009: 88 for the combinations), but they are predictable for weak verbs. Ending formation includes suffixation as well as vowel alternations in the stem/root (metaphony) and associated alternation in consonant quality of the root-final consonant (i.e., palatal vs. non-palatal); insertion of u into the root/stem is called u-infection (Stifter 2009: 67).[22] Apart from ‘normal’ active endings, there are separate inflectional endings known as deponent, also conveying an active meaning, constituting a ‘merely lexical property that has to be known for

[20] W, S and H are according to the classification system by McCone (1997); Thurneysen’s (1946) AI and AII are equivalent to McCone’s (1997) W1 and W2, respectively.

[21] e.g., **cér* ‘I bought’, **cíuir* ‘he bought’.

[22] Examples of this can be found with prs. ind. and prt. (active) 1sg. conj. inflections with root *ber*: *·biur*, *·tabur*, *·biurt*, *·tuburt*. Cf. the paradigms for *beirid* and *do·beir* in Appendix B, page 162 and pages 166–167, respectively.

each verb separately' (Stifter 2009: 87). As mentioned in section 2.2.1, there is a distinction between absolute and conjunct endings, enforced by morphosyntax.

Strachan (1949) and Green (1995) are convenient reference guides for examples of all the verb types and their endings, as well as for paradigms for frequent strong and irregular[23] verbs.

2.3 Stem allomorphy

I have attempted to illustrate the key features causing stem allomorphy in section 2.2.1 and section 2.2.3. Accentual patterns are integral to verb stem formation in Old Irish. The rigid stress system of Old Irish results in 'complex synchronic morphophonemic alternations' (Stifter 2009: 90) and, consequently, a system of 'double stem formation' (Russell 2005: 431). According to Stifter (2009: 60):

> The bewildering complexities [...] become transparent only when viewed from a diachronic position, and in order to understand allomorphic variation correctly it is essential to work with underlying forms and their often quite dissimilar surface representations.

Another key observation by Stifter (2009: 85) is the following:

> Verbs display a striking propensity towards compounding with up to four preverbs before the root. [...] Semantic information being thus shifted from the roots towards the preverbs, the role of the roots has been enervated in consequence. This is on the one hand reflected in the fact that in many synchronic stem allomorphs the roots are no longer visible or are heavily truncated. On the other hand, a diachronic result of this is the reduction of the number of inherited roots [...] and the high proportion of compound verbs in relation to simple verbs.[24]

Allomorphic variation is most prevalent in compounds, which can be said to have two stems, a deuterotonic and prototonic one (cf. section 2.2.1). Unlike deuterotonic forms, which have 'a kind of barrier or juncture across which certain otherwise normal processes do not occur', prototonic forms have their first preverb 'fully incorporated into the rest of the verb' (McCone 1997: 4). The possible forms for one inflectional form for a simplex and compound in the preterite paradigm are given in Table 2.8. The verbs both have root *lēc* and are weak, resulting in predictable stem consonants by means of suffixation in the tense/mood combinations. The prt. 3sg. form, however, is anomalous in that the absolute ending does not have an ending suffix, while neither a preterite stem consonant (*-s*) nor an ending suffix is present in conjunct inflection. Note, finally, that the compound *do·léici* only consists of one preverb, so stem alternation is still reasonably transparent in the light of the underlying form of the preverb and verb root. Another compound with the same verb root *lēc* but with two preverbs is

[23] Verbs with suppletion in some of their tense/mood paradigms.

[24] The observation pertaining to the reduction of inherited roots is made in Wodtko (2007).

(prs. ind. 3sg.) *as·oilgi*, dependent/prototonic *·oslaici* ‘opens’ (*uss-od-lēc-* according to Stifter (2006: 364)), with a higher discrepancy between root forms and surface stems. As stated in section 2.2.1, compound verbs may have up to four preverbs; the more preverbs, the more inflectional variation.

Table 2.8 – Stem allomorphy illustrated with prt. 3sg. forms of the verbs *léicid* (lēc-) ‘lets’ and *do·léici* (to-lēc-) ‘lets go’. Bold = lexical elements, italics = stressed. E = ending.

Form	Roots	Schema	Dependency	Verb type
(1) ***léic***is	lēc-	VROOT $\mathrm{E}_{\mathrm{ABS}}$	independent	simple
(2) ní·***léic***	ní-lēc-	CONJ_PART · VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	simple
(3) ro·***léic***	ro-lēc-	AUG · VROOT $\mathrm{E}_{\mathrm{CONJ}}$	independent	augmented simple
(4) ní·*rei***lic**	ní-ro-lēc-	CONJ_PART · AUG VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	augmented simple
(1) **do**·***léic***	to-lēc-	PV · VROOT $\mathrm{E}_{\mathrm{CONJ}}$	independent	compound
(2) **do**·*rei***lic**	to-ro-lēc-	PV · AUG VROOT $\mathrm{E}_{\mathrm{CONJ}}$	independent	augmented compound
(3) ní·***tei*lic**	ní-to-lēc-	CONJ_PART · PV VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	compound
(4) ní·***tar*laic**	ní-to-ro-lēc-	CONJ_PART · PV AUG VROOT $\mathrm{E}_{\mathrm{CONJ}}$	dependent	augmented compound

Put in a very general way, the Old Irish verb has three parts which are subject to huge variability: the beginning, the middle, and the end. The beginning is usually a preverb or particle, the middle is the stem, and the final part the ending. As I showed in sections 2.2.4, 2.2.5 and 2.2.6, the picture becomes more complex if the augment or affixes such as infixed pronouns and emphasising particles are added. On top of that there are the consonant mutations (sometimes caused by morphemes that are only underlyingly present) that are often not marked in the orthography, but have a grammatical significance. Although proclitic prefixes and a two-way system of endings (absolute vs. conjunct) add significantly to the inflectional complexity of verbs, their formations are relatively transparent in terms of the underlying morphemes; Chapter 4 will show that it is possible to derive most of the prefixes and suffixes in the verbal complex by rules pertaining to individually defined morphemes.

The real challenge, it was found, lies in defining the surface shape of composite elements constituting the ‘middle part’: the part between the proclitic juncture and the ending. The somewhat unpredictable nature of this kind of ‘stem’ variation, especially from a synchronic linguistic viewpoint, is, as mentioned above, ultimately due to the stress system. The unpredictability effectively results from the position to which the composite parts of the verbal complex (preverbs, augment, verb root) happen to be assigned. It follows then, that the highly variable inflectional variation equals unstressed verb root: the stress shifting to the left relative to the verb root (forms (4) of *léicid* and (2)-(4) of *do·léici* in Table 2.8).[25]

[25] Suppletion, stems supplied from different roots, adds to the abundant stem variation in Old Irish.

2.4 Quantifying inflectional variation

2.4.1 Monolithic stems

In this thesis I am operating with two notions of verb stem:

1. The traditional five bases in Old Irish (McCone 1997): the present, subjunctive, future, preterite active and preterite passive stem. This could be termed the tense/mood stem.

2. The 'middle part' of the verbal complex, i.e., all components in the verbal complex from the stressed syllable up until and including the verb root.

Notion 2., which I call a *monolithic stem*, is best illustrated with simplified schemas of the verbal complex. Compare the elements constituting the monolithic stem in Table 2.9 with the accompanying verbal forms in Table 2.8.

Table 2.9 – Monolithic stems marked in the Early Irish verbal complex schemas. E = ending.

Schema	Dependency	Verb type
$\boxed{\text{VROOT}}\ \text{E}_{\text{ABS}}$	independent	simple
CONJ_PART · $\boxed{\text{VROOT}}\ \text{E}_{\text{CONJ}}$	dependent	simple
AUG · $\boxed{\text{VROOT}}\ \text{E}_{\text{CONJ}}$	independent	augmented simple
CONJ_PART · $\boxed{\text{AUG VROOT}}\ \text{E}_{\text{CONJ}}$	dependent	augmented simple
PV_1 · $\boxed{\text{PV* (AUG) PV* VROOT}}\ \text{E}_{\text{CONJ}}$	independent	compound
CONJ_PART · $\boxed{\text{PV* (AUG) PV* VROOT}}\ \text{E}_{\text{CONJ}}$	dependent	compound

The monolithic stem can be thought of as a unit of which the constituting parts combine in such a way that the result may not be trivially segmentable on the surface, i.e., a base unit that is treated as if it was not being composed of individual morphemes (preverbs, augment, verb root). Note that the pretonic preverb in independent (deuterotonic) compounds—obviously a lexical element—is not regarded as part of the monolithic stem. This is not theoretically motivated, but rather due to the way deuterotonic compounds are encoded in the computational architecture (cf. Chapter 4). The concept of a monolithic stem translates into a workable definition for computational purposes. This does not mean that the underlying morphological structure will be lost during the implementation. The computational paradigm of two-level morphology (cf. section 3.3.3.2 and section 4.2), used in this thesis to create a morphological parser for Old Irish, is well suited for retaining access to underlying roots at all times.

The monolithic stem concept is also interesting from a more theoretical point of view. With monolithic verbal stems one can derive all attested Old Irish verbal forms with simple morphological rules. When a substantial number of verbs have been encoded, the minimum average of verbal stems needed can be calculated, which gives one a quantifiable diagnostic for the complexity of the Old Irish verbal system. Moreover, type and distribution patterns of these invariant units might emerge that reveal something about the underlying rules.[26]

Encoding complex stem formation by means of monolithic stems is in principle independent of the distinction between weak and strong verbs; both with weak and strong verbs, and particularly with the compounds within both types, many 'non-transparent' formations arise that need to be 'hard-coded' using these invariant bases (cf. Table 2.8). Weak and strong verbs do, of course, behave rather differently in terms of the first notion of stem: weak verbs show predictable stem formation by means of suffixation, while strong verbs show stem formation through suffixation, vowel alternations (ablaut) and reduplication.[27]

Provided that the monolithic bases have been identified, and that syncope is correctly applied, the inflectional pattern across a weak verb's paradigm is almost entirely predictable, although one gets analogical formations too (cf. section 4.6.4.4). For most simple weak verbs, this comes down to at most two monolithic stems (non-augmented and augmented). For most weak compound verbs, a minimum of four bases are needed (deuterotonic, prototonic, and the augmented versions for each).[28] Although the concept of monolithic stem entails that one has to define and operate with more than one stem for most verbs, it is my intuition that the computational encoding of these non-derived monolithic units outweighs by far the method of formulating and implementing what must be a vast amount of often idiosyncratic derivational rules from abstract (and potentially reconstructed) underlying roots, especially with strong verbs.

2.4.2 Average size of the verbal paradigm

Old Irish has eight different tenses and/or moods: present indicative, past habitual, present and past subjunctive, future and secondary future (conditional), preterite and imperative. In most tenses/moods, two inflectional variants exist (independent and dependent), across three persons, singular and plural (accumulating to six), plus two passives (singular and plural). This results in 8 x 2 or 16 different forms. Simple verbs have an additional 5 (passive and non-passive) relative forms in the present indicative, present subjunctive, future and preterite. In the other tenses/moods there are only seven or eight forms, invariably consisting of forms with conjunct endings (past habitual, past subjunctive, conditional and imperative). This amounts to 115 forms with simple verbs. Compound verbs do not have separate relative endings, but have independent and dependent (deuterotonic and prototonic) forms in all tenses/moods, amounting to 8 x 16 or 128 forms. One can thus work with an average of 120 inflectional forms for verbs in general.

These figures are obviously an underestimation of the total amount of forms in the Old Irish verbal system, since orthographical by-forms, preverbal particles (including augments

[26] I am indebted to Prof. David Stifter for bringing these additional insights to my attention.

[27] Compare the verb paradigms provided in Appendix B. As mentioned in section 2.2.7, ablaut may be the result of reduplication, as is the case with root *crī* in the verb *crenaid*, for which cf. pages 164–165 in Appendix B.

[28] Choices made in relation to the computational encoding of stem entries have resulted in listing augmented stems of simple verbs under their (simple verb) lemma, rather than creating a separate classification system or treating them as (lexical) compounds. For the rationale of operating on a lemma basis, cf. section 4.7.3.

to create perfective forms) and affixes (infixed pronouns, for example), which are part of the verbal complex, are not included. When those are included, the figure of about 120 inflectional forms per verb increases exponentially.[29]Appendix B shows the full paradigms for a selection of strong and weak verbs.

In addition to the large amount of inflectional forms, varying scribal and editorial practices cause different kinds of markers to be employed in texts to separate proclitic elements from the stressed part of verbal complex. Section 4.4.2 in the implementation chapter delves into challenges relating to spacing (or, rather, the lack thereof) within the verbal complex and the formulation of associated morphosyntactic dependencies. Some of these issues have also come up during testing; cf. section 5.6.1.

2.4.3 Available lists of verbs and verb roots[30]

The focus in my work, as stated in section 2.4.4, is on weak verbs. In order to get some grip on the amount of (computational) work that is needed, is it possible to quantify the balance between weak and strong verbs? In other words, how many are there of each, both in an absolute sense and relative to one another?

An early publication listing verb roots is Pedersen (1909–13), who lists 204 roots.[31] Unfortunately, however, the majority of roots given are primary verbs, a closed class of verbs with Proto-Indo-European roots, which tend to be strong verbs in Old Irish. In other words, Pedersen (1909–13) left out the largely denominative weak verbs, presumably as their inflection is more predictable, even though he states that (following translation mine):

> [Z]u den Unregelmäßigkeiten der Flexion kommt im Ir. noch die vom Präverbsystem und von der Enklise des Verbums veranlaßte Unregelmäßigkeit, sodaß schließlich die allermeisten der häutiger vorkommenden Verba unregelmäßig sind (Pedersen 1909–13: Vol. 2, 449).
>
> In addition to the irregular inflection, Old Irish exhibits a system of preverbs and irregularity caused by enclisis, so that in the end most of the more commonly occurring verbs are irregular.

Pedersen's concept of irregularity, while unhelpful—irregular on what level: suppletion, apparent discrepancy between abstract root shape and stem, 'wrong syncope', etc.?—resonates with the observations made in section 2.3, in the sense that stem formation is complex both with

[29] Arguably, one could add conjunctions of the type *co* 'until' or *má* 'if', constituting a consecutive string with the rest of the verbal complex (at least in manuscripts). In contrast, standardised Modern Irish has 'only' 34 forms across seven tense/mood paradigms (it lost the past subjunctive and preterite; the past originates from the perfect), translating into an average of about five distinct forms in each tense/mood paradigm. This reduction in forms is partly due to historical 3sg. forms being employed as analytic forms in the various tenses/moods.

[30] This section is partly based on the *Proceedings of the Thurneysen Fanclub*: issue 40; records of the discussions in the Conference Room on 21/05/2018 in the context of the ERC-funded project *Chronologicon Hibernicum*, Maynooth University, Ireland. Proceedings are available at `http://nuim.academia.edu/RudolfThurneysen`.

[31] Based on the dedicated number of paragraphs: 650–854.

weak and strong verbs (a justification for the employment of monolithic stems, section 2.4.1). As stated by Prof. David Stifter in the proceedings on which this overview is partly based, however, 'most verbs are regular, but with devilishly complex rules governing their surface forms'. I would rather use the term 'synchronic stem formation predictability', i.e., predictable vs. (much) less predictable. Thurneysen (1946: § 857) gives a selection of verbs with their deuterotonic and prototonic forms.

A recent work on primary verbs in Celtic is Schumacher (2004), who lists 166 strong verbs for Irish, as can be seen in Table 2.10. There is one caveat in this work, however: causatives were not included. Collections of secondary (mostly weak) verbs, with a more 'predictable' inflection, are rare. Le Mair (2011) has collected a good number of them. Her corpus consists of the Old Irish glosses. She counted 365 weak verbs, with 106 W1 verbs (97 non-deponent), and 259 W2 verbs (64 non-deponent).[32] Le Mair (2011) also collected and described the primary verbs, but gives no statistics.

Table 2.10 – Root count based on Schumacher's 2004 *Die keltischen Primärverben* (the totals signify the amount of roots listed in Schumacher's work). *One instance with **k*- only attested in Middle Irish: *scibid* 'fließen, schwimmen (von einem Wasserfahrzeug), (sich) ruckartig bewegen' (Schumacher 2004: 423).

Initial sound	Irish	Total	Initial sound	Irish	Total
*a-	7	7	*k$^{\text{u̯}}$-	4	5
*ā-	2	2	*l-	15	17
*b-	11	14	*m-	9	12
*d-	12	15	*n-	3	3
*e-	3	3	*o-	1	1
*g-	10	11	*φ-	7	10
*g$^{\text{u̯}}$-	4	4	*r-	11	13
*i-	1	2	*s-	26	29
*ī-	1	1	*t-	13	17
*i̯-	1	1	*u-	1	1
*k-	12*	15	*u̯	12	14
			Grand total	**166**	**197**

There are other publications listing verbs, although none of these have aimed at a complete list categorised according to stem class. The authoritative eDIL (cf. Appendix A, section A.1.1.1) comprises of 4,127 verb entries. However, not all verbs are accompanied by stem class.[33] Rossiter (2004) applied the stem class classification system of McCone (1997) (the one adhered to in the present work) to verbs in eDIL, but, unfortunately, only dealt with compounds. The *index verborum* in McCone (1997) lists various inflections for each verb, but does

[32] Le Mair (2011) used the classification by Thurneysen (1946), with AII encompassing both W2a and W2b.

[33] An example is *caraid* 'loves', which is (largely) W1.

not include stem class and is based primarily on the glosses. The vocabulary section in Stifter (2006) is not exhaustive but does give the stem classes.

2.4.4 Focus in the present work

Weak verbs are the focus of my thesis due to (in most cases) predictable patterns of stem consonant suffixation (e.g., an *f*-future) and accompanying endings. I have limited myself to the most frequent weak verb categories, W1 and W2a. The W2 subclass W2b, consisting of causatives with fluctuation of the root vowel,[34] is rather small. The full paradigm of W1 *marbaid* 'kills' and W2a *léicid* 'lets' are found in Appendix B, on pages 173–174 and page 172, respectively. These verbs are also important for the implementation, cf. Chapter 4.

As I chose to build a morphological parser from the ground up, any additional complications in relation to stem variation, the most challenging aspect of the verbal system, were avoided. In other words, in the light of both time constraints and additional layers of complexity, it seemed justified to ignore both W2b and strong verbs for the moment. In laying down the computational morphological infrastructure, however, I have anticipated and catered for the inclusion of strong (and W2b) verbs, exactly because my solution for handling stem variation already revolves around the input of more than one (monolithic) stem for a verb, whether weak, strong, simplex or compound.

A focus on weak verbs is also justified in light of the fact that weak verb inflection becomes the dominant type of verb formation in Middle Irish (the generalisation of the *f*-future for example, one of the two main future formations surviving in the modern language). Deponent inflection has also been ignored as it is gradually phased out during the Middle Irish period, with former deponent verbs assuming active endings (Russell 1995: 55). A good example of this is *molaid* 'praises', the headword in eDIL; deponent *molaithir* would be more representative of Old Irish.

The irregular verbs of Old Irish are those that show suppletion in their paradigms; these are equally not dealt with.[35] Some observations relating to stem entry issues with verbs with root *ber* will be discussed in section 4.7.3.

2.5 The verb in Middle Irish, and beyond

McCone (1997) discusses the developments in the verbal system in Middle Irish in great detail in his last chapter 'Key Middle-Irish developments'. Broadly speaking, three developments can be identified:

[34] An example of a member of this class is *roithid*, 'makes run, sets in motion', alternatively *ruithid*, as mentioned in eDIL (`http://dil.ie/35498`).

[35] The paradigms of the strong verbs *beirid* and *do·beir* in Appendix B show suppletion with perfective *ro·ucc-* in the case of *beirid* (p. 163) and perfective *do·rat-/do·ucc-* in the case of *do·beir* (pages 168–169). Apart from these suppletive stems, one would need to define at least the monolithic stems *bér*, *bert* and *breth* and prototonic (for *do·beir*) *tabair*, *tibér*, *tubart*.

- Development of an immutable root shape and transparent stem formation.
- The replacement of pronominal affixes by independent object pronouns.
- Homogenisation of personal endings.

The strategy to arrive at an immutable root shape was primarily to employ prototonic compound bases as a template for new simple verbs, based on analogy with dependent forms of old simple verbs (McCone 1997: 192), e.g., *léicid* : *·léici*, X : *·teilci*, where X = new simple verb = *teilcid* (< Old Irish *do·léici*). This phenomenon can be described as univerbation, a lexicalisation process involving the 'unification [...] of a syntactic phrase or construction into a single word' (Brinton & Traugott 2005: 48). This conversion is most evident when a former compound occurs with absolute inflection. In parallel with compound verbs becoming (weak) simplexes, the augment assumes the status of conjunct particle (which is invariably unstressed) with the 'virtual elimination of the oppositions between unaugmented and augmented forms' (McCone 1997: 165), reducing the allomorphic stem variation. Compare the Old Irish form in (16), the augmented variant of prt. 3sg. *do·léic*, with the Middle Irish (simple verb) equivalent in (17) (with invariant stem *teilc*).

(16) **do-rei-lic**
to-ro-lēc
PV-AUG-let.PRT.IND.3SG
'has let go'

(17) **ro-teilc**
AUG-let.go.PRT.IND.3SG
'(has) let go'

The preverbal particle *ro* is generalised as an augment in Middle Irish (Breatnach 1994: 279, McCone 1997: 187) and is replaced by *do* in the later language, which is also the standard form in Classical Modern Irish (McManus 1994: 408). However, in dependent position *ro* survives (*níro* > *níor*). Note that contemporary past tense forms derive from augmented preterite (or perfect) forms, not from 'bare' preterites. According to Williams (1994: 455), the preterite disappears from the spoken language by the beginning of the 17th century. The *s*-preterite 3sg. was still common, albeit only in absolute form, in the writings of the Early Modern Irish writer Geoffrey Keating (Bergin 1930: xxii).

Due to final unstressed vowels becoming indistinct in Middle Irish (Russell 2006: 990), ambiguity arose with different endings in the verbal paradigm. This resulted, for example, in the generalisation of the zero-ending in prs. ind. 3sg. conj. taken from strong verbs of the type *benaid*, *·ben* 'hits' (McCone 1997: 205), such as *·marb*, *·léic* instead of *·marba*, *·léici*, respectively. Analogical processes were also involved in a subsequent Middle Irish development whereby the prs. ind. conj. ending *-ann* / *-enn* (modern *-ann* / *-eann*) became prevalent, originally restricted to specific verbs (McCone 1997: 206–207). Between the end of the Early

Irish period and the beginning of the Early Modern Irish period one can observe the rise of independent subject pronouns, paving the way for a movement from synthetic to analytic verb forms, with the employment of a 3sg. as the generic inflectional form (Greene 1958, Greene 1973; cf. also examples given in Breatnach 1994).

Table 2.11 shows the development of some verbs between Old and Modern Irish using eDIL (prs. ind. 3sg.), Dinneen (1927) (prs. ind. 1sg.) and Ó Dónaill (1977) (*FGB*, imp. 2sg.).[36] The Middle Irish forms marked with an asterisk have a separate entry in eDIL.[37] Most verbs in this list are compounds of which the prototonic form supplied the template for the new simple verb in Middle Irish. In the case of *in(d)-fét*, the Early Irish verbal noun *indisiu*, *indisin* provided the new base. Middle Irish *at-beir* 'says' contains a fossilised infixed pronoun neuter *-t* (McCone 1997: 172), which was incorporated into the stem (with *t* > *d*). The fossilised infixed pronoun also accounts for the evolution *at·rubart* > *a·tubart* > *(a)dubhairt* > *dúirt* 'said' (McCone 1997: 172, 190–191, 204).

Table 2.11 – Verb roots/lemmas between Old and Modern Irish.

Root(s)	eDIL	Middle Irish	Dinneen	*FGB*	Translation (infinitive)
to-ad-ell	*do-aidlea*	*taidlid**	*taidhlim*	*tadhaill*	'to approach, touch'
ess-ber	*as-beir*	*at·beir*	*adeirim*	*abair*	'to say'
to-lēc	*do-léci*	*teilcid*	*teilgim*	*teilg*	'to cast'
dī-slond	*do-sluindi*	*díltai(gi)d*	*diúltuighim*	*diúltaigh*	'to deny'
fo-gab	*fo-gaib, -geib*	*fag(b)aid*	*do-gheibhim / faghaim*	*faigh*	'to get'
inde-fēd	*in(d)-fét*	*indisid**	*innisim*	*inis*	'to tell'
lēc	*léicid*	*léicid*	*leigim*	*lig*	'to let'
marb	*marbaid*	*marbaid*	*marbhuighim*	*maraigh*	'to kill'

2.6 Synthesis

The main goal of this chapter was to point out—leaning on excellent literature on the subject—how the phonology of Old Irish imposes itself upon the morphology. The interface between morphology and phonology is most complex in the verbal system. The main skeleton of the verbal complex, as well as various means of affixation, have been illustrated in section 2.2. Extremely important for Chapter 4 are factors that cause a significant degree of stem allomorphy (section 2.3), notably the distinction between independent and dependent, leading to divergent bases most clearly seen with compounds (i.e., deuterotonic vs. prototonic). These alternations are ultimately due to stress system of Old Irish, with syncope often causing heavy truncation of the verb root.

[36] Strictly speaking, forms are not directly comparable due to variation in tense, mood and person.

[37] Although only a cross-reference to *do·aidlea* in the case of *taidlid*.

In section 2.4 I have attempted to quantify inflectional variation, which includes a workable definition of a base for Old Irish verbs, which I call a monolithic stem. The focus in the present work is on weak verbs which show predictable suffixation in their 'tense/mood stems' (e.g., an *f*-future). Some tentative numbers relating the amount of strong and weak verbs, based on non-exhaustive lists, have been provided to get a rough indication of the coverage of my project.

An overview of the main changes within the Early Irish period—with reference to Modern Irish forms—has been provided in section 2.5.

Chapter 3

Computational approaches and methodology

3.1 Introduction

Using computational approaches to deal with language variation in historical texts is far from straightforward. Piotrowski (2012: 9) has observed that ‘there is no underlying computational model that describes how synchronic and diachronic variants relate to each other and—possibly—to some shared meaning or some kind of prototype that represents the relatedness of the variants’. Piotrowski (2012) documents an extensive amount of literature on the subject. The projects described exhibit a multitude of approaches, guided by various linguistic subdisciplines and, most importantly, by the needs and the characteristics of the historical language in question. In other words, there is no ‘one size fits all’ approach in computational linguistics for historical texts, and, unfortunately, Old Irish is no exception. Old Irish too poses language-specific computational challenges (cf. Chapter 4) due to its complex morphology, the sparseness of data (even if one aggregated all texts in archives such as CELT and TLH, cf. Appendix A, section A.1.2) and the disparate and discontinuous nature of the various projects and resources, already pointed out in section 1.5.

This chapter is structured as follows. In section 3.2, I will introduce the field of computational linguistics by giving a historical overview and explaining the key terms and concepts. Sections 3.3, 3.4 and 3.5 discuss important concepts and linguistic subtasks in Natural Language Processing. Section 3.6 looks at projects and approaches dealing with parsing historical language forms and texts. This is not straightforward: does one categorise the efforts according to language or language typology, linguistic subdiscipline targeted (e.g., orthography, phonology, morphology, morphosyntax, syntax), main method of the project (morphological analysis, Part-Of-Speech Tagging, lemmatisation), goal and audience (providing access to cultural heritage, facilitating students, etc.) or computational paradigm used (rule-based vs. statistical)? Moreover, the projects that will be discussed employ—in an often unique fashion—pre-existing or newly created lexical resources, and the quality and comprehensiveness of these lexical resources ultimately determine the nature and combination of approaches employed in the project (as well as its limitations). In surveying the various projects, I have made an undoubtedly arbitrary categorisation, prioritising method and linguistic subdiscipline targeted: orthographical standardisation (section 3.6.1), morphological analysis and lemmatisation (section 3.6.2), and Part-Of-Speech Tagging (section 3.6.3). My own methodology is laid out in section 3.7 while section 3.8 synthesises the matters discussed in this chapter.

3.2 Background to computational linguistics

Speech and Language Processing (SLP) is concerned with the ability of computers to process human language (Jurafsky & Martin 2009: 35). SLP is an interdisciplinary field, almost as old as the computer itself, and was until relatively recently characterised by diverging frameworks and practices, reflected by the variety of disciplines that contribute to the field: computer science, linguistics, electrical engineering and psychology/cognitive science (Jurafsky & Martin 2009: 25, 43). The historical development of the field must be understood in terms of these contributing disciplines, as well as the theories and models that the main players adhered to.

The different historical paradigms and approaches have given rise to different names for the field, the most common of which are SLP, Natural Language Processing (NLP), Computational Linguistics and Human Language Technology. Computational Linguistics and NLP are—arguably—the most common names for the discipline. The term Computational Linguistics used to be associated with formal language theory (part of the so-called symbolic paradigm, cf. below) (Kay 2003) and was associated with linguistics departments, while NLP derives from a computer science context (Jurafsky & Martin 2009: 43).

NLP appears to be preferred when the focus is less on a linguistic framework, and more on engineering problems. Thus, in the preface to *Handbook of Natural Language Processing*, it is stated that 'the emphasis is on practical tools', that the handbook is 'aimed at language-engineering professionals', that it strongly focuses 'on the *how* of the techniques rather than the *what*' and that '[s]uch a focus also continues to distinguish the handbook from recently published handbooks on *Computational Linguistics*' (Indurkhya & Damerau 2010: xxi). However, in defining Computational Linguistics, The Association for Computational Linguistics states that:

> Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system.[1]

Thus, the fields overlap to such an extent that a distinction is hardly relevant. In the present

[1] `https://www.aclweb.org/archive/misc/what.html`.

thesis, therefore, the terms Computational Linguistics and NLP will be used interchangeably, without signifying a paradigm or theory.

Two different models exist in Computational Linguistics. The first one is knowledge-based ('hand-crafted'), the second is data-driven ('statistical' or 'empirical'),[2] a distinction that echoes the development of the two main research paradigms during the 1950s and 1960s, which Jurafsky & Martin (2009: 44–45) call the symbolic and the stochastic paradigm, respectively. Whereas the symbolic paradigm is guided by formal language theory as defined by linguists such as Chomsky and others, the stochastic paradigm is associated with electronic engineering centers and their statistical approach to, initially, automatic text recognition, and later (1970s), automatic speech recognition and synthesis.

In subsequent decades, a proliferation of paradigms and methods can be observed, with the field coming together in the late 1990s. By this stage, probabilistic methods have been incorporated in domains which were previously dominated by a rule-based or 'hand-crafted' approach, including parsing (cf. section 3.3) and Part-Of-Speech Tagging (cf. section 3.3). The introduction of the World Wide Web in the 1990s is crucial in that it accelerates research into language-based information retrieval and extraction. The 2000s are characterised by the availability of growing amounts of spoken and written data and the rise of machine learning (cf. section 3.4), which is largely based on statistical methods.

3.3 Linguistic tasks and the NLP pipeline

3.3.1 Linguistic annotation

A central concept in computational text processing is *linguistic annotation*. Linguistic annotations are 'notes about linguistic features of the annotated text that give information about the words and sentences of the text [...] which can be used by subsequent applications (Wilcock 2009: 1). For example, words labelled with morphological information can be subsequently lemmatised (more on which below). Important for the purposes of the present work is annotation on the orthographical and morphological level. Linguistic annotations are obtained by what is often referred to as *parsing* (although without a qualifier the term generally refers to syntactic parsing): taking an input form and producing a structured linguistic representation (Jurafsky & Martin 2009: 79). In the remainder of this section I will describe the various activities in the Natural Language Processing pipeline for the linguistic subdisciplines.

3.3.2 Pre-processing and orthography

The first step is pre-processing, which includes tokenization: separating marks and other non-orthographical characters from words (Jurafsky & Martin 2009: 167). If spelling adheres to a standard next in line is morphological analysis. If not, the next step is spelling normalisation. In its broadest sense this subtask aims to arrive at a consistent spelling, which constitutes

[2] https://www.aclweb.org/archive/misc/what.html.

the fundament of lexical resources, statistical methods and information retrieval (Piotrowski 2012). In historical text processing, one encounters both synchronic (e.g., dialectal, stylistic) and diachronic variation. A canonical form may be a common, normalised or hypothesised historical spelling, but it may also be a modernised orthographical form. In the latter case, normalisation is more accurately described as spelling modernisation (Piotrowski 2012: 70). Jurish (2010: 72) defines a *canonical cognate* as a modern form that preserves 'both the root(s) and morphosyntactic features of the associated historical form(s)'.

Spelling variation may be dealt with using techniques of *approximate matching*, determining the similarity between between two strings (Jurafsky & Martin 2009: 107–108). Approximate matching is well-explored in Information Retrieval as well as widely used in spell checkers (Piotrowski 2012). An important metric of similarity is the Levenshtein distance, which aligns two strings and calculates the minimum number of editing operations (insertion, deletion, substitution) needed to transform one string into another, often with particular costs assigned to each of these operations. Figure 3.1 illustrates the Levenshtein distance with Classical Old Irish *teilcem* (dependent prs. ind. 1pl.) and Middle Irish *tilgem* 'we let go, cast', etc.

```
T * I L G E M
| | | | | | |
T E I L C E M
  i     s
```

Figure 3.1 – Minimum edit distance with Old and Middle Irish cognates. The Levenshtein distance is either 2 (insertion (i) and substitution (s)), or, alternatively, 3, if substitution is assigned a cost of 2, i.e., a combination of deletion and insertion (Jurafsky & Martin 2009: 108).

3.3.3 Morphology

3.3.3.1 Morphological parsing, stemming and lemmatisation

When the problem entails recognising strings on the word-level one speaks of morphological parsing. The linguistic subdiscipline of morphology deals with morphemes, 'the minimal linguistic units with a lexical or grammatical meaning' (Booij 2012: 8–9). Two broad classes of morphemes can be identified: the 'main morphemes' (stems) and 'additional meanings' (affixes) (Jurafsky & Martin 2009: 81). These morpheme classes are otherwise known as free or lexical morphemes and bound morphemes, respectively (Booij 2012: 9). The procedure of automatic stemming reduces inflected forms to their root or stem (Jurafsky & Martin 2009: 80). Lemmatisation is a related task, except that the 'common denominator' between two or more strings needs to be found; one wants to group inflected words under its base form (Mitkov 2005: 744). This base form might be more abstract than a stem, or based on a conventional citation form, as found in dictionaries. Old Irish is a good example of a language where the root, stem and lemma of a word may be completely different.[3]

3.3.3.2 Finite-State Transducers (FSTs)

State machines, or automata, recognise a particular set of symbol sequences (strings) as defined by a regular expression.[4] Automata can be conceptualised as networks with transitions through a finite amount of paths. Finite-State Transducers (FSTs) are finite-state automata with two-level relations for each path in the network. These inherently bidirectional mappings are very well suited for linguistic modelling, especially morphology, employing the notion of a lexical and surface level. Figure 3.2 visualises an FST as a network, mapping the surface string `léicid` (prs. ind. 3sg., 'lets') to the lexical string `lēc +VROOT +PRS +IND +3P +SG` (and vice versa). FSTs are the subject of section 4.2.

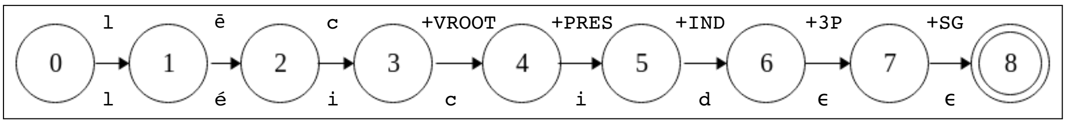


Figure 3.2 – A Finite-State Transducer (FST) accepting, at final state 8, a set of two-level symbol mappings: `lēc+VROOT+PRS+IND+3P+SG:léicid` (`lexical : surface`). The epsilon (ϵ) denotes a so-called 'empty transition': a mapping where there is no accompanying symbol on the opposite level, i.e., when the upper and lower strings are of unequal length. 'Analysis' is used for upward mapping, which translates into morphological parsing. Downward mapping equals 'generation' of (commonly) orthographical strings.

FSTs constitute a well-established computational paradigm extremely well-suited—yet surprisingly little used with historical languages—for the modelling of morphology. Jurafsky & Martin (2009: 80) describe an FST as a 'key algorithm for morphological parsing [...] and crucial technology throughout speech and language processing'. A foundational work on two-level morphology is Koskenniemi (1983).

3.3.4 Part-Of-Speech Tagging

Part-Of-Speech Tagging (POS Tagging, or just 'tagging') constitutes the link between morphology and syntax (constituent structure and word order). POS Tagging involves assigning syntactic class makers to each word in a corpus, resolving ambiguity on the word level (Jurafsky & Martin 2009: 167). The list of POS Tags employed for a corpus is a fixed set, called the tagset, defining (or taking over from the morphological analyser output) the categories such as verb and noun and their accompanying features (singular or plural for nouns, tense and mood for verbs, etc.) (Wilcock 2009: 27–28).

POS-tagging and lemmatisation are common activities in NLP for historical languages; the creation of a rule-based morphological parser less so. This is possibly due to the fact that building a morphological parser from the ground up is labour-intensive, and can be circumvented

[3] For example, the compound with the citation form (prs. ind. 3sg.) *do·léici* 'lets go' consists of the root elements *to* and *lēc* and one of its stems, namely aug. prt. 3sg., is *tarlaic-*.

[4] A language for specifying text search strings (Jurafsky & Martin 2009: Chapter 2).

if the linguistic distance between the historical and modern variety is ‘bridgeable’ by using a modern-language POS Tagger and an orthographical standardisation/modernisation module (Piotrowski 2012: 87). The Classical Language Toolkit[5] (Kyle P. Johnson 2014–2017) offers NLP support for the languages of Ancient, Classical, and Medieval Eurasia. Latin and Greek are served best in the Toolkit; for both languages there is a lemmatiser and a POS Tagger available.

3.3.5 Parsing on the sentence level

In a subsequent step, the aim is to arrive at a representation of constituent and sentence structure, referred to as syntactic parsing, commonly shortened to *parsing*. A treebank is a parsed corpus with syntactic annotation. Parsing might also refer to semantic parsing. The current work does not deal with syntactic or semantic parsing.

3.4 Rule-based vs. machine-learning methods

In current NLP research, insights from linguistics only partly inform the computational strategies involved, and language-independent (often statistical) methods complement manual rules. If one wants to normalise historical texts, for example, one can manually define orthographical rules as mentioned in grammars, etc. However, it is also possible to use language-independent and unsupervised string distance methods (for ‘unsupervised’ cf. below). Similarly, POS Tagging may be rule-based; disambiguation rules need to be formulated that specify constituent structure. An example of such a rule is that a determiner is not usually followed by a verb. Most modern-day taggers use a (complementary) machine-learning component such as statistical methods based on the likelihood of certain words (tags) occurring together, usually based on a manually created and/or adapted training corpus (Jurafsky & Martin 2009: 169).

Machine learning is a field interested in improving performance by making accurate predictions on the basis of a data set (Mohri, Rostamizadeh & Talwalkar 2012). This involves learning functions that map a set of input numbers to an output number (Kelleher 2016). Predictions are often made by statistical inference. Statistical methods are widely employed in machine translation, which uses a target language and a translation model based on conditional probabilities (Koehn 2010). The last few years have seen the rise of deep learning or neural network approaches and their increased usage in machine translation. Kelleher (2016) provides a short overview of neural machine translation for a non-expert audience.

The machine-learning paradigm makes an important distinction between ‘supervised learning’ and ‘unsupervised learning’ (Mohri, Rostamizadeh & Talwalkar 2012). In supervised learning, the ‘experience’ is in the form of labelled data (e.g., morphological analysis or POS Tags), while unsupervised methods take as input ‘raw’, unlabelled data.

[5] `http://cltk.org/`.

3.5 Testing

Finally, two concepts in relation to corpora and testing need to be explained. A ‘gold standard’ corpus is a test set out of a corpus that has been annotated (grammatically parsed) and checked by a human annotator. This test set can be used to evaluate the (automatic) tagger accuracy, based on percent correct, which is 96% to 97% for simple tagsets (Jurafsky & Martin 2009: 189). The percentage reported on is often an F-score (Jurafsky & Martin 2009: 479), the weighted harmonic mean between *precision* and *recall*. Precision measures the ratio between correctly analysed forms out of the (possibly limited) forms retrieved. Recall, on the other hand, specifies how many correctly analysed forms were selected out of the total of relevant items.

3.6 Computational approaches to historical texts

3.6.1 Orthographical standardisation

There are various methodologies available to deal with spelling variation. In the words of Borin & Forsberg (2011: 42):

> Which approach is chosen for any particular case may of course vary depending on the availability of language resources and tools for the modern language, linguistic expertise in the research group, and whether the knowledge residing in the language tools is primarily in the form of manually formulated rules or statistical, acquired through machine learning.

Etxeberria et al. (2016) identify three canonicalisation techniques: rule-based methods (hand-written phonological grammars), machine-learning (statistical) techniques using standard-variant pairs, and unsupervised methods, e.g., phonetic distance and edit distance (cf. section 3.3). They themselves use a (semi-)supervised machine-learning method; they use Phonetisaurus, a Weighted FST driven phonology tool, to learn mappings of phonological changes using a noisy channel model, and apply this method to Basque, Spanish and Slovene texts, resulting in F-scores above 80% in the case of Basque.

Dereza (2016) has developed an Early Irish Lemmatiser using form-lemma mappings extracted from eDIL. The tool consists of a lemma predictor which employs the so-called Damerau-Levenshtein distance, checking for all possible strings of the forms on edit distance 1 and 2 (cf. the Figure 3.1 for an example). In other words, the tool predicts a mapping between an unknown orthographical variant and a known one, extracted from eDIL.[6] It subsequently returns the eDIL headword.

Dereza (2016) compiled a corpus of c. 100,000 tokens from 24 thematically related, mainly Early Irish, texts published on Corpus of Electronic Texts (cf. Appendix A, section A.1.2.1).

[6] Strictly speaking, the forms in eDIL are grammatical inflections and not necessarily orthographical variants; however, in the XML-markup they are individually tagged as <oVar>.

The Lemmatiser shows a 76.31% average recall score (cf. section 3.4) and is able to predict lemmas for out-of-vocabulary words.[7] The use of the rule-based Early Irish Lemmatiser as an ancillary resource is reported on in a case study using the text *Táin Bó Fraích* (cf. section 5.7). Section 6.4 shows how the latter is envisaged to be part of a larger linking framework for historical cognate verb forms.

VARD2[8] is a standardiser and web interface to deal with Early Modern English texts, based on spell-checking (edit distance) measures (Baron & Rayson 2008). The automatic discovery of extant canonical cognates for historical German (Jurish 2010) is also based on string distance (Levenshtein). Bollmann, Petran & Dipper (2014) combine machine-learning techniques and edit distance methods for a rule-based approach to modernising Early New High German text. They align the 1545 Luther bible with its modernised version, deriving mapping rules similar to phonological rewrite rules by recording the edit operations and the left and right context. Probabilities of generated modern forms are calculated in case of multiple output variants, and checked against a dictionary to establish whether the form exists. The rule-based approach was compared against a word list substitution approach, i.e., substituting a historical form with a modern word form that it is most often aligned with. A combination of the two approaches yields the best results (93%), but performance drops to 42% with more diverse language data. The normalisation tool Norma was developed in the context of standardising Early New High German and can be adapted to different varieties of historical data (Bollmann 2012).

For pre-standard 'Revival' Irish[9] (1882-1926) texts, Uí Dhonnchadha et al. (2014) report on lemmatisation and POS Tagging of a 7-million-word corpus of Irish which has been modernised using a lexical database of historical and modern word pairs, together with supervised statistical machine learning as well as rule-based techniques. A standardiser (*An Caighdeánaitheoir*)[10] developed by Scannell (2008) is employed in conjunction with a modern-language POS Tagger (Uí Dhonnchadha & van Genabith 2006), the backbone of which is a morphological FST. The three components of *An Caighdeánaitheoir* are the following:

1. *Manual rewrite rules*. These consist of hand-written orthographical rewrite rules of the type *sg-* ⟶ *sc-*.

2. *Machine translation* (text alignment). Although a 100-million-word web corpus of texts published after the spelling and grammar reform (post 1958) was available, none of these texts were found to entirely conform to the modern standard. This problem was resolved by using rule-based grammar and spelling correction[11] and automated standardisations

[7] `https://github.com/ancatmara/early-irish-lemmatizer`. Dereza subsequently developed a lemmatiser using neural networks, or deep learning, which may prove useful in future work.

[8] http://ucrel.lancs.ac.uk/vard/about/

[9] Cf. section 1.2.1 on the historical stages of Irish.

[10] Cf. `http://cs.slu.edu/~scannell/pub/acis17-paipear.pdf` for a short overview of standardisation for contemporary Modern Irish.

[11] On the basis of proofing tools devised by Kevin Scannell: *An Gramadóir* and *Gaelspell*, cf. `http://cs.slu.edu/~scannell/gaeilge.html`.

to a small number of recurring words, to create a sub-corpus of 40 million words approximating the standard language (Uí Dhonnchadha et al. 2014: 14-15).

The translation model assigns the same conditional probability to mappings from the lexical database. The conditional probability of non-standard/standard pairs following from the invocation of rewrite rules is calculated by 'penalising' method, which involves multiplying the application of each rule with a fixed factor (< 1). Decoding proceeds from left-to-right using a trigram language model.

The body of parallel ('bilingual') texts available in both pre-standard and standardised Irish is small (700,000 words) with significant variation in pre-standard texts, resulting in noisy alignments, which, as such, are not suitable for the high accuracy translation task at hand. Consequently, the translation model was defined differently, using a more suitable hybrid approach (Uí Dhonnchadha et al. 2014), employing the manual rewrite rules in 1.

3. *Expert knowledge* provided by Irish linguists specialising in the historical periods. To date, this consists of 22,000 pre-standard lemmas mapped to their standard forms, plus an additional 10,000 variants taken directly from the Ó Dónaill (1977) dictionary.

Initial calculations point to F-scores ranging from 91-96%, while POS-Tagging accuracy is 89%.

String similarity can also be measured by using a statistical method based on n-grams (consecutive elements): if certain letters in a word tend to be followed by the same letters in other words, those words might be variants of the same form. This method has been employed for 16th–18th-century English texts (Robertson & Willett 1992) and Medieval French from the 12th century (O'Rourke et al. 1997).

Finally, the problem of historical spelling variation can be tackled by abstracting from spelling and use letter-to-sound correspondences (a phonetic representation), provided that these correspondences still exist in the modern language. Thus, by using a tool or algorithm (for example a text-to-speech system) for the modern language, older spellings can be mapped to a phonetic representation, which can be used to retrieve modern spelling. Jurish (2008) used this approach to produce modern, canonical cognates from orthographical variants in historical German texts.

3.6.2 Morphological analysis and lemmatisation

Rule-based morphological analysis never occurs as an isolated method—one needs a list of stems as contained in a dictionary, for example, as well as morphological rules (typically taken from a grammar). Early work was limited to the ancient Indo-European languages—Greek, Latin and Sanskrit. Packard (1973) is one of the earliest efforts in applying computational morphological analysis to historical text for classical Greek, a highly inflected language. The goal of the project was to facilitate first-year university students by providing them with a method of instruction based on features occurring in texts—as opposed to a more 'abstract' year-long course before being exposed to a significant quantity of literature.

The program by Packard (1973) incrementally strips off final letters until it recognises the string as an inflectional ending. Ancient Greek allows multiple prepositional prefixes and an augment before the verb root. Prepositional prefixes have assimilated forms depending on the following consonant. The program by Packard tries to parse a word as a stem + ending; when this fails, an algorithm tries to strip off a hypothetical prefix, arrives at a prefix-stem division (sometimes more than one) and, when analysis is successful, reunites prefix and stem.

Smith (2016) reports that he has not found a published automatic parsing algorithm that succeeds on ancient Greek. The latter's parser architecture includes an FST, but the interrelation of morphological accent, syllabic quantity and movable accent cannot be modelled using a sequential series of transducers. The algorithm employed involves 'analysis by synthesis' methodology, which, according to Smith (2016), is similar to the way in which the *Morpheus* tool[12] (Crane 1991) works. First, accents are stripped off and the accent-free token is analysed using an FST. The various accentual possibilities are then algorithmically applied and compared to the original input, accepting the matching form(s) and rejecting the ones accented differently.

Passarotti (2010) reports on the existence of three morphological analysis tools for Latin: LEMLAT,[13] Whitaker's *Words*,[14] and *Morpheus*, which was originally created for ancient Greek (Crane 1991) and is now integrated in the Perseus Project.[15]

Huet (2003, 2005) reports on segmentation and morphological analysis for Sanskrit. The lexicon contains inflected forms generated by internal *sandhi*[16] (word-internally, that is, across morphemes) from a stem dictionary annotated with grammatical information. External sandhi processes (across word boundaries, which are less complex and local) are modelled using FSTs. The last step consists of sentence segmentation employing the inverse of external sandhi phenomena.

Morphological rewrite rules were used as an aid to improve lemmatisation in the context of the XML-encoded *Dictionnaire du Moyen Français* (*DMF*)[17] for Middle French (c. 1330–c. 1500). The tool *Lemmes, Graphies et Règles Morphologiques* (Souvay & Pierrel 2009) consists of a morphological component that augments the collection of lemmatised spellings and known lemmas to hypothesise the modern lemma from any given form. The morphological rewrite rules are accompanied by a precondition (optional) and a post-condition, specifying position in the word, and POS, respectively. When tested on a corpus text of the DMF from 1410, Souvay and Pierrel (2009) found that in 60% of the cases the lemmatiser produces one lemma, which is the correct one. In 39% of the cases it produces many lemmas, including the correct one.

[12] A rule-based program with a database of 40,000 stems, 13,000 inflections, and 2,500 irregular forms.

[13] `http://www.ilc.cnr.it/lemlat/`.

[14] `http://archives.nd.edu/words.html`.

[15] `http://www.perseus.tufts.edu/hopper/`.

[16] sandhi means 'joining' in Sanskrit, i.e., the phonological modifications happening when forms are joined to one another (Parodi & McCarthy 2010: 372-373)

[17] `http://www.atilf.fr/dmf/`.

Borin & Forsberg (2011) report on SALDO (Swedish Associative Thesaurus version 2),[18] a lexical-semantic tool enriched by POS Tags and a morphological component. SALDO functions as the pivot resource between modern Swedish and a diachronic lexical resource for historical stages of Swedish. As Borin & Forsberg (2011) point out, Old Swedish is characterised by significant spelling variation and rapid language change. During the second half of the Old Swedish period the language underwent a development from the Old Norse mainly synthetic type to the present largely analytic type, and the sound system was thoroughly reorganised.

A Late Modern Swedish (1733–1906) dictionary served as the basis for the generation of inflectional information for that language period, with 80% of the verb entries being covered. For the morphological component of SALDO the framework of Functional Morphology is employed, a tool that provides a development environment for computational morphologies (Forsberg & Ranta 2004, Forsberg 2007).

The lexical tool for Old Swedish (1225-1526)—a historical variety considerably removed from Contemporary Modern Swedish—is based on three historical dictionaries and manually extracted inflectional patterns by an expert, resulting in 3,000 lexical entries being provided with inflectional information, again implemented by using Functional Morphology. The considerable linguistic variation was handled computationally by treating ending variation in the morphological component and stem variation as a spelling problem (at the time of writing edit distance or other string similarity measures were being considered). While linking the lexical resources for late Modern Swedish and Contemporary Swedish is relatively straightforward, Borin & Forsberg (2011) have not yet found a working solution for linking these resources with Old Swedish.

The 5,000 conjugated Old Irish verb forms in the online lexical resource *In Dúil Bélrai* (cf. Appendix A, section A.1.1.3) constitute partial lemmatisation tables and are integrated in *Wordlink*, which links webpages word-by-word to online dictionaries, and in *Multidict*, a multiple dictionary lookup facility (Ó Donnaíle 2014).[19] Multidict incorporates a headword suggestion mechanism based on lemmatisation tables and algorithms, which can be prioritised in different ways. It also facilitates linking to eDIL. The 5,000 conjugated verb forms could be input into the FST framework presented in Chapter 4. Another resource developed by Caoimhín P. Ó Donnaíle is *Bunadas*.[20] This is a Celtic cognates network database using a clustering mechanism to encode relationships between etymologically related words. However, support for the Middle and Early Modern period is limited at the moment. In this thesis lemmatisation will therefore be experimented with using Dereza (2016).

3.6.3 Part-Of-Speech Tagging (mainly statistical)

To be able to deal with historical texts, one can decide to either create a POS tagger ‘from scratch’, or adapt a ‘modern-language’ tagger. When resources such as annotated corpora

[18] `https://spraakbanken.gu.se/eng/resource/saldo`.

[19] Both tools are are available at `http://multidict.net/`.

[20] Available at `https://www2.smo.uhi.ac.uk/gaidhlig/faclair/bunadas/`. Cf. also Appendix A.1.1.4.

(for statistical tagging) or modern-language taggers are lacking, creating a POS Tagger ‘from scratch’ is the only real option. This is normally restricted to ancient or ancestral languages, which are too far removed from the modern language (if a modern variant exists, of course). It is therefore hardly surprising that POS Taggers were specifically created for ancestral languages or early varieties of a language (Piotrowski 2012): Latin (Passarotti 2010), Classical Chinese (Huang et al. 2002) and Old French (Stein 2007). This approach is used for a Latin statistical POS Tagger, based on manually annotated corpora in the context of PROIEL,[21] the Perseus Latin dependency treebank[22], and the *Index Thomisticus Treebank* (Passarotti 2010).

Lynn (2012) used the Python modules of the Natural Language Toolkit (Bird, Klein & Loper 2009)[23] to show how computational methods could be applied to medieval Irish texts. Part of her experiments include a rudimentary POS-tagged version of the Old Irish text *Táin Bó Fraích* ‘the cattle-raid of Fróech’, edited by Meid (1974).[24] This text will serve as testing ground for my Old Irish FST in Chapter 5 and the lemmatiser developed by Dereza (2016).

Rögnvaldsson & Helgadóttir (2011) used a statistical POS Tagger previously trained on Modern Icelandic corpus to deal with 13th and 14th-century Old Norse saga texts. However, Old Norse is still relatively close grammatically to Modern Icelandic and the text editions were in Modern Icelandic orthography. Even with a 700-item tagset, after manual correction and unioning the ‘old’ and ’modern’ training corpora, an accuracy level of 92.7% was reached.

A somewhat more common approach, however, is to adapt a modern-language tagger (if, available, of course) by an orthographical standardisation module (rule-based or statistical). A corpus can be modernised (cf. section 3.6.1) before it is input to a POS Tagger, generally producing between 80% and somewhat over 90% accuracy rates. Examples of this approach include Rayson et al. (2007) for Early Modern English, Scheible et al. (2011) for Early Modern German and Uí Dhonnchadha et al. (2014) for ‘Revival Irish’ (cf. section 3.6.1).

If a significantly sized parallel corpus is available (typically old and modern bible editions), one can use a method known as ‘bootstrapping’: projecting the modern tags onto the old text by text alignment, and then training a POS Tagger on this annotated old text. Moon & Baldridge (2007) used this approach to train a POS Tagger on Middle English biblical texts.

[21] Pragmatic Resources in Old Indo-European Languages, a research project aiming at a close linguistic study of the language in the Greek text of the New Testament as well as its translations into the old Indo-European languages Latin, Gothic, Armenian and Old Church Slavonic. It has created a treebank of ancient Indo-European languages, including Latin and Ancient Greek, cf. https://www.hf.uio.no/ifikk/english/research/projects/proiel/ and https://proiel.github.io/.

[22] https://perseusdl.github.io/treebank_data/.

[23] The original book for Python 2, as well as an updated version for Python 3, is available at http://www.nltk.org/book/. The toolkit is available at http://www.nltk.org/.

[24] Available at Corpus of Electronic Texts, http://www.ucc.ie/celt/published/G301006/.

3.7 Methodology employed in the present work[25]

The methodology employed in my project bears most resemblance to approaches focusing on morphological analysis and lemmatisation, cf. section 3.6.2. As mentioned in sections 3.3 and 3.6.3, for ancestral and ancient languages, the linguistic distance between the old and the modern variety (if the older variety survives as a modern language, of course) restricts the use of a modern-language POS Tagger or other 'modern' resources (if existing at all). Old Irish constitutes a language phase too remote, linguistically speaking, from Modern Irish. In other words, seeing that no automatic morphological parser for Old Irish exists, creating one was deemed necessary. Both the modern, contemporary standard and Classical Old Irish (8th and 9th centuries) can be treated as normative phases in the history of the language, and are well resourced. As already discussed in section 1.6.1, the language of the Old Irish glosses (Classical Old Irish) constitutes the basis for Old Irish grammars and is used to assess texts of the later medieval period. In other words, starting on automatic morphological parsing for Old Irish is justified on many grounds.

In my project I have operated with the conceptual methodology illustrated in Figure 3.3. It must be emphasised, however, that many of the constituting parts in this framework are outside the remit of my thesis. The approach is fundamentally based on a 'two-pronged attack': one arrow reaching forward and the other one reaching back. Two automatic morphosyntactic parsing tools at the opposite end of the chronological spectrum are envisaged, covering the historical period of Irish. The backbone of both tools is a morphological FST. As much progress has been made on automatic parsing of historical texts for the Modern Irish period (Uí Dhonnchadha et al. 2014, Mac Cárthaigh 2018), work was started on the Early Irish period, of which Old Irish is the most stable variety and much better resourced compared to Middle Irish. This is an important reason for focusing on this period in my project.

The idea is that a verb form (or any other word) will receive a parse—via bidirectional standardisation—either in Old or Modern Irish using the FSTs. Ideally one arrives at a full morphological analysis for Old Irish, but, failing that, a form should be lemmatised using the lemmatiser developed by Dereza (2016) for Early Irish—arriving at the eDIL headword, most of which are Middle Irish with some Early Modern. The method of standardisation and tagging is most fully worked for Modern Irish. The output of the *Caighdeánaitheoir* developed by Scannell (2008) for pre-standard Modern Irish is successfully piped to the Modern Irish tagger developed by (Uí Dhonnchadha & van Genabith 2006), which contains the lemmas of *FGB* = Ó Dónaill 1977.

The bidirectional adaptation process has not been completed for either Early or Modern Irish, although Uí Dhonnchadha et al. (2014) report on very good recognition results for *Corpas Stairiúil na Gaeilge* (1600–2000) (section 3.6.1). A corpus of Bardic Poetry from roughly the 1200–1650 period (cf. Appendix A, section A.1.2.3) has been subjected to the morphological FST and POS Tagger for modern contemporary Irish, also by using the standardiser developed

[25] This section is based on section 7.2.1 in Fransen (forthcoming).

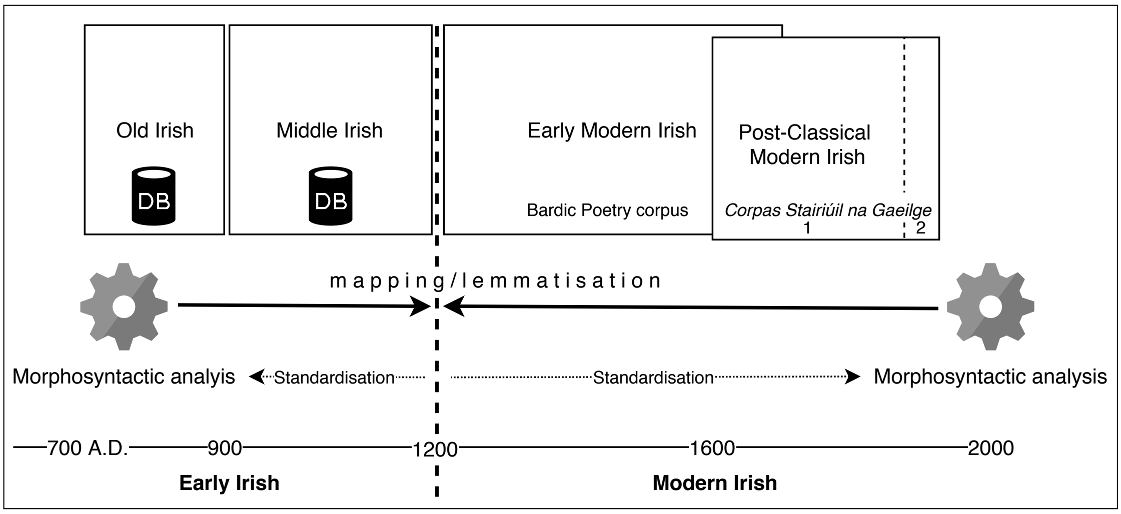


Figure 3.3 – A 'two-pronged attack' to map cognate verb forms in Irish.

by Scannell (2008), giving promising results.[26] In this thesis, the focus is on the finite-state implementation of a subset of verbs for Old Irish (Chapter 4), with an attempt to facilitate analysis and generation of forms adhering to normalised Old Irish.

Rather than solving all the issues relating to the linking and adaptation processes in Figure 3.3, which is a vast amount of work, I set out to explore ways of linking and mapping historical cognate verb forms in this thesis. Anticipating further advancement of the adaptation processes in the near future, I will propose a mapping architecture in Section 6.4. The proposed architecture for Early Irish is to be understood in terms of three key points presented below, the first one of which is most fully worked out in this thesis and constitutes the most substantial part of the work (cf. Chapter 4).

1. The creation of a rule-based morphological parser, using a Finite-State Transducer (FST), for Old Irish verbs.

2. Incorporation of manually parsed verb forms from the dictionaries/databases *Chronologicon Hibernicum* and *In Dúil Bélrai* into the FST.

3. Employing lemmatisation and standardisation methods for Early Irish, based on Dereza (2016), in conjunction with morphological analysis for Old Irish.

Automatic morphological analysis of Old Irish verbs has proven to be a challenging undertaking, even when meaningfully restricting the scope to the weak verb classes W1 and W2a. This work therefore does not deal with parsing beyond the morphological level; however, the current work paves the way for developing a POS Tagger which is able to successfully recognise verb forms using morphosyntactic disambiguation strategies. I will return to this matter in section 4.4.2.

[26] Dr Eoin Mac Cárthaigh, presentation as part of the Bardic Poetry Workshop, held on 12/05/2017 at Trinity College Dublin, cf. `https://bardicpoetryworkshop.wordpress.com`.

The automatic morphological parser, developed as part of this work, aims to generate normalised or 'standard' forms, approximating Classical Old Irish grammatical and orthographical features. This allows for the possibility of incorporating the contents of—or at least testing the implementation against—the manually parsed (verb) forms in the Old Irish Glosses databases (cf. section 1.5 and Appendix A, section A.1.1.2), representing Classical Old Irish. Incorporation of this material is very much future work, as various databases are currently being streamlined in the context of the *Chronologicon Hibernicum* project.[27] Verb forms from these databases could be extracted and imported into the morphological FST architecture—with adaptation and streamlining of the tag systems of both tools. Work on a POS Tagger for Old Irish is planned for the immediate future within the *Chronologicon Hibernicum* project, based on previous work in relation to POMIC (cf. section 1.5). Possibilities for collaboration between my project and *Chronologicon Hibernicum* are currently being investigated.

Combining morphological analysis (and POS Tagging) with spelling normalisation, similar to the *Foclóir Stairiúil na Gaeilge* 'the Historical Dictionary of Irish' project (1600–2000),[28] was initially planned as part of my project, but was not feasible due to the unexpected complexities of building a morphological parser for Old Irish verbs. Chapter 5 will explore and test the added value of lemmatisation (Dereza 2016), discussed in section 3.6.1, in the context of a case study using the text *Táin Bó Fraích*, which contains some Middle Irish forms. Section 6.4.4 discusses the prospect of employing Dereza's 2016 Early Irish Lemmatiser for normalisation purposes. Creating separate FSTs to deal with variants and unknown forms is the subject of section 6.4.5.

3.8 Synthesis

This chapter has introduced the field of Computational Linguistics or Natural Language Processing (section 3.2), the NLP pipeline (section 3.3) and showed how computational techniques can be employed for historical texts (section 3.6). The focus of this overview has been on orthography, morphology and lemmatisation, which constitute the most important areas of my project. Due to factors such as language typology, data sparseness, availability of resources and linguistic distance between historical and modern variety, computational techniques are seldom transferable to other historical languages. Based on the projects surveyed, it seems that the creation of rule-based, linguistically-informed morphological parsing tools specifically for historical languages is uncommon and is restricted to classical languages, notably Greek and Sanskrit.

A more common approach—when a modern/standard variety exists—is to utilise a POS Tagger for that variety, in conjunction with a spelling modernisation component, which might include statistical alignment methods (as with Post-Classical and later Modern Irish texts, Uí

[27] Introducing the *Chronologicon Hibernicum*. Paper presented at the 10th Celtic Linguistics Conference (CLC10), 4-5 September 2018,`https://www.maynoothuniversity.ie/sites/default/files/assets/document/Celtic%20Linguistics%20Conference%20-%20Abstract%20Booklet_2.pdf`.

[28] Cf. section 1.5 and *Corpus Stairiúil na Gaeilge* in Appendix A, section A.1.2.4.

Dhonnchadha et al. 2014). Common methods for lemmatisation include encoding morphological or orthographical rules and approximate string matching techniques to arrive at headwords. The Early Irish Lemmatiser (Dereza 2016) uses an approximate matching algorithm to arrive at an Early Irish headword in eDIL. This Lemmatiser is relevant for matters discussed in Chapters 5 and 6.

Due to the lack of resources for Early Modern Irish, the linguistic distance between Old and Modern Irish, and the fact that Old Irish has received much (digital) scholarly attention, it was decided to start work on an automatic morphological Finite-State Transducer (FST) for Old Irish. This FST will be instrumental in a bidirectional adaptation approach or 'two-pronged attack' (Figure 3.3) with morphosyntactic parsing tools and lemmatisation tools at the opposite ends of the (historical) chronological spectrum, representing the most comprehensively resourced Irish language periods. The Old Irish morphological FST will be used in conjunction with a lemmatiser based on eDIL (Dereza 2016), which, after having been augmented with generated forms from the Old Irish transducer, can be used as a standardiser (section 3.7). A more detailed, yet preliminary, framework for creating mappings between Old and Modern Irish verb forms will be introduced in Chapter 6.

Chapter 4

Implementation

4.1 Introduction

This chapter deals with the core objective of the thesis: building a morphological parser for Old Irish verbs. The computational paradigm employed is based on Finite-State Transducers (FSTs), which are the topic of section 4.2. The instruments (software) used, accompanying coding conventions and the test set of verbs employed are described in section 4.3. Section 4.4 lists the choices and challenges in the context of modelling the Old Irish verbal complex. The implementation is based on two stand-alone digitally encoded lexicons, one for unstressed proclitic elements ('prefixes'), and one for stems and endings, which are discussed in section 4.5 and section 4.6, respectively. Operationalising morphotactic restrictions and (separated) dependencies is an important theme in this chapter, especially in relation to stem entries for compound verbs. Section 4.7 is devoted to implementing these non-trivial morphological processes. The encoding of more general morphotactic restrictions when combining the two stand-alone lexicons mentioned above is the subject of section 4.8. Section 4.11 provides a synthesis of the most important points covered in this chapter.

This chapter contains code snippets that illustrate the workings of the FST. Each example is accompanied by the name of the file and a page reference to the relevant section in Appendix C, which lists all code files. The line numbers in the code excerpts match the ones in the original files from which they were extracted. The code is also available online.[1]

4.2 Two-level finite-state machines: transducers

A finite-state automaton or machine (FSA) is a model that recognises a particular set of sequences of symbols (or strings) as defined by a regular expression. A regular expression is a metalanguage formulated in algebraic notation for characterising a set of strings (Jurafsky & Martin 2009: 51–52). An example of a regular expression is `a+`, meaning one or more `a`'s (`a`, `aa`, `aaa`, etc.) Regular expressions and FSA's represent exactly the same set of languages

[1] `https://github.com/ThFransen84`.

called regular languages (Hopcroft, Motwani & Ullman 2001: Chapter 3), and are thus mathematically equivalent. A regular expression compiles into a finite-state network, which encodes a (possibly infinite) language (Beesley & Karttunen 2003: 44).

FSAs are designed to model operations that can be characterised by a finite number of steps, each resulting in a different state. The machine can be in only one state at a time, and there are a finite number of states to which it can proceed. A finite-state network has a start state and a final (or accepting) state, which are not necessarily different. A change of one state to the other—a transition—is triggered by a condition or event, and is graphically represented by an arc. In finite-state networks, each symbol results in a transition from one state to the next, producing a path through the network. Individual symbols can stand for anything, but in modelling natural languages they often denote morphemes, phonemes or orthographical characters. Traversing through the various paths constitutes the accepted symbol combinations of the machine. The legal strings contained in the FSA define its language. In other words, if an input string matches a path in the network then it is a valid string in the language, otherwise it is rejected, i.e., not part of the language of the machine. The FSA corresponding to `a+` is given in Figure 4.1.

The aim is to make the FSA contain only legal or desired strings; in linguistics this exercise reflects building a correct grammar of a language, often with a focus on morphology, phonology or orthography. In the present work, which deals with Old Irish verbs, the finite-state model incorporates both morphological, phonological and orthographical features. The current implementation uses an extension of an FSA, known as a Finite-State Transducer (FST) or lexical transducer, which constitutes the more commonly used paradigm in modelling a language's morphology.

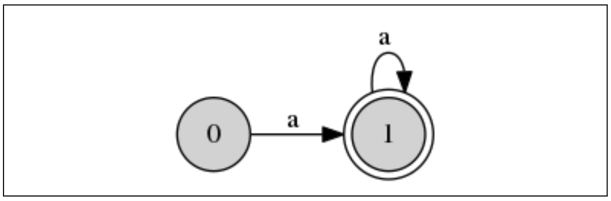


Figure 4.1 – The regular expression `a+` represented as a finite-state automaton. State 1 is the final state, marked with a double circle.

Such an FST translates (transduces) a lexical level-symbol into a surface-level symbol. The lexical level is often represented as the upper layer, while surface (e.g., orthographical) strings constitute the lower level. This convention is adhered to in the present work. The set of ordered pairs of strings in a two-level system is known as a relation. Analysis or look-up refers to the process whereby lower-level symbols are consumed and upper-level symbols produced, whereas during generation or look-down upper-level strings are consumed, giving surface forms (Beesley & Karttunen 2003: 9–14). Figure 4.2 shows an example of a transducer for a regular relation that translates the symbol a (lexical level) into b (surface level), and vice versa. The power of an FST is that it is inherently bidirectional.

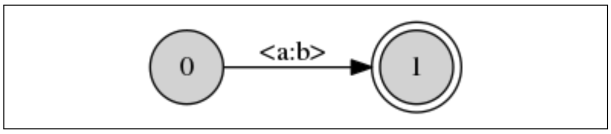


Figure 4.2 – The regular relation <a:b> visualised as a finite-state network.

Antworth (1991) gives an overview of the similarities and differences between two-level phonology and (early) generative phonology. Although transducers are intricately linked with two-level models of morphology and phonology developed from classical rewrite rules as codified in Chomsky and Halle's 1957 publication on generative phonology, *The Sound Pattern of English*, generative rewrite rules create dynamic changes resulting in intermediate derivational forms that have no access to either the lexical level nor the surface form. A two-level model, however, is characterised by static correspondences between lexical and surface symbols. In other words, unlike in classical generative phonology, lexical or underlying symbols remain available (and can be evoked) in subsequent two-level rules, exactly because of the fact that paired symbols are encoded as a relation, with the correspondences being static.

According to Beesley & Karttunen (2003: 33), the ground work was already laid in 1972 when Johnson theorised that phonological rewrite rules could be modelled as Finite-State Transducers. Johnson (1972) was also right in claiming that through rule composition any cascade of transducers can be represented by a single transducer. As Jurafsky & Martin (2009: 114) have pointed out, Johnson's insight was independently discovered by Kaplan and Kay and published in their 1981 article *Phonological rules and Finite-State Transducers*. Kaplan and Kay's work was subsequently followed up and most fully worked out by Koskenniemi (1983), who successfully applied two-level morphology to Finnish, a highly agglutinative language.

The mathematical possibility of combining an arbitrary cascade of alternation rules with intermediary forms into one single FST is visualised in Figure 4.3. Beesley & Karttunen (2003: 36) illustrate how a rule transducer, when combined with a lexicon of underlying/abstract forms, results in a single all-inclusive network, known as a *lexical transducer*, as shown in Figure 4.4. The output of my work described in this chapter is a lexical transducer for a subset of Old Irish verbs: rule information and a lexicon in a single data structure.

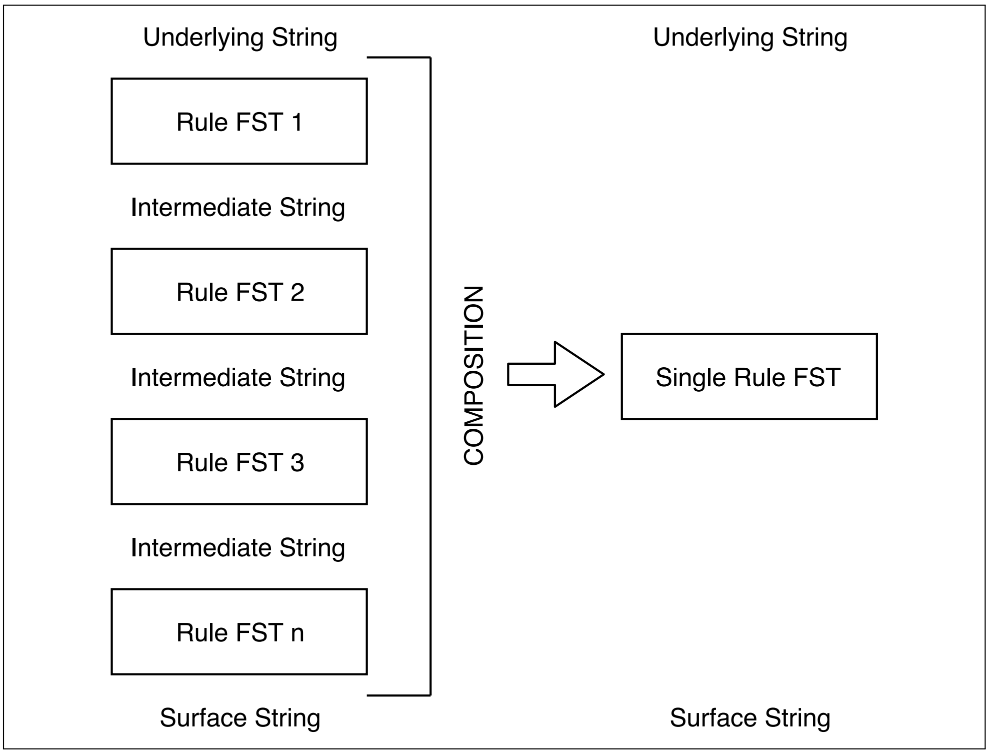


Figure 4.3 – Combining a cascade of alternation rules with intermediary forms into one Finite-State Transducer. Taken from Beesley & Karttunen (2003: 35).

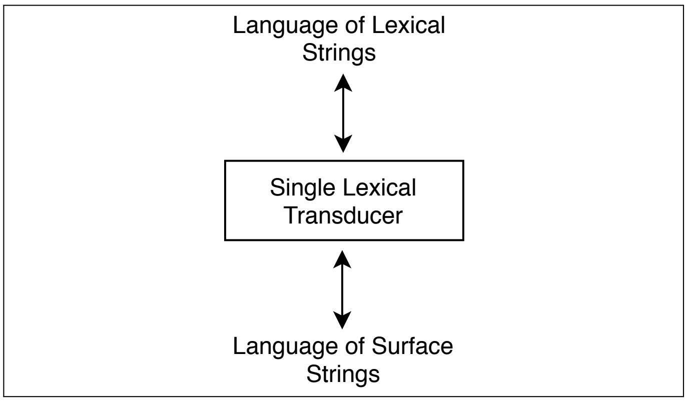

Language of Lexical Strings
Single Lexical Transducer
Language of Surface Strings

Figure 4.4 – A single all-inclusive or lexical transducer. Taken from Beesley & Karttunen (2003: 36).

4.3 Inventory of the FST toolkit

4.3.1 Instruments: finite-state tools used

There are a few tools for finite-state computing that are freely available. These include HFST (*Helsinki Finite-State Transducer Technology*)[2] and SFST (*Stuttgart Finite State Transducer*)[3]. With OpenFst[4] one can attribute weights to transitions in a finite-state network, representing the costs of taking a particular transition. The finite-state toolkit foma (Hulden 2009)[5] provides additional support for first-order regular logic expressions and includes functions for restraining reduplication. The latter is used in the present work. It is an (augmented) non-licensed reimplementation of the licensed Xerox-tools,[6] which were developed in the 1990s-2000s. The core program of the Xerox-tools is xfst (*Xerox Finite State Transducer*). The Xerox-tools are accompanied by Beesley & Karttunen (2003),[7] an extremely well-written and accessible companion on finite-state morphological modelling of natural languages. Nonetheless, foma was chosen for the current implementation, not only since it does not come with a license, but also because I established with Mans Hulden,[8] developer of foma, that there is a bug in xfst in relation to the elimination algorithm for *flag diacritics* (cf. section 4.3.5), which are heavily used in the present work. It must be stated that while foma claims to be compatible with the Xerox-tools, it was found that there are a few minor non-compatible differences. Moreover, foma is somewhat less intuitive than the Xerox-tools in communicating parsing errors during compilation. A hugely beneficial addition in foma, relative to xfst, however, is that only a single command is needed to eliminate all flag diacritics from the network.

4.3.2 The lexc format: building lexicons

The lexc (lexicon compiler) program (Beesley & Karttunen 2003: Chapter 4) facilitates intuitive encoding of lexicons which are also easy to maintain. In the Xerox-tools there is a designated stand-alone program which can also be invoked in xfst; in foma, the lexc functionalities are integrated in the main compiler. lexc files (which I accompany with the .lexc suffix in the implementation) are invoked as illustrated in Code Example 4.1, with optional command elements in brackets. The lexicon file will be interpreted and put on the stack,[9] which now contains +1 network.

[2] Available at http://hfst.sourceforge.net/hfst3/index.html.

[3] Available at http://www.cis.uni-muenchen.de/~schmid/tools/SFST/.

[4] Available at http://www.openfst.org/twiki/bin/view/FST/WebHome.

[5] Availableathttps://fomafst.github.io/.

[6] https://web.stanford.edu/~laurik/.book2software/.

[7] The website of the book is https://web.stanford.edu/~laurik/fsmbook/home.html.

[8] Via email communication, 07/10/2017.

[9] A limited-access ordered data structure of elements defined by the user.

Code Example 4.1 – Invoking a `lexc` file in `foma`, which is subsequently compiled and put on the stack.

```
foma[0]: (read) lexc (<) file.lexc
foma[1]:
```

The basic architecture of a `lexc` file is shown in Code Example 4.2. A `LEXICON` represents a morpheme type (or letter, phoneme, etc.) and contains forms specified in a two-level relation. Each form or relation, e.g., <`a:b`>, is assigned a *continuation class* which leads to a sub-lexicon. The transducer compiled from the lexicon in Code Example 4.2 contains the (valid) string relations <`ab:de`> (<`a:d`> followed by <`b:e`>) and <`abc:def`> (<`a:d` followed by <`b:e`> followed by <`c:f`>). The relation <`a:d`> may stand for <`lēc+VROOT:léic`>, with the lexical (or upper) level on the left of the colon, and the surface (or lower) level on the right. This can also be visualised as $\frac{\texttt{l ē c +VROOT}}{\texttt{l é i c}}$. If the optional morpheme <`c:f`> stands for <`+EMPH+3P+SG+FEM:-si`>, the longest string in the transducer, <`abc:def`>, might stand for (18) or, equivalently, (19).

(18) <`lēc+VROOT+W2a+PRS+IND+ABS+3P+SG+EMPH+3P+SG+FEM:léicid-si`>

(19) $\frac{\texttt{l ē c +VROOT +W2a +PRS +IND +ABS +3P +SG +EMPH +3P +SG +FEM}}{\texttt{l é i c i d - s i}}$

Code Example 4.2 – Basic structure of a lexicon in `lexc` format.

```
LEXICON X
a:d     Y; ! <- continuation class

LEXICON Y
b:e     Z; ! <- continuation class

LEXICON Z
        #; ! concatenation stops here
c:f     #; ! <- optional morpheme, e.g., a suffix
```

4.3.3 Regular expression operators

As detailed in section 4.2, a regular expression and a finite-state automaton are two sides of the same coin. Compiling regular expressions in `xfst` and `foma` is facilitated by a regular-expression metalanguage, which differs somewhat from standard formalisms. Important operators, commonly used in the present work, are given in Table 4.1. They are found in Beesley & Karttunen (2003: 45–54, 84–97); the examples are geared towards my own implementation. Square brackets (`[ ]`) are not assigned a semantic interpretation; they are used syntactically to give precedence to operators (and can be conveniently and redundantly used to make the code more legible).

Table 4.1 – The main regular expression operators used (the preceding command `regex` is not shown).

Operator	Name	Usage example	Example description
`{ }`	Concatenation	`{tá}`	Equivalent to `[t á]` (not to be used with multicharacter symbols).
`->`	Replacement	`' -> 0`	Rewrite palatalisation marker into the empty string (i.e, delete). For empty string cf. next operator.
`0` *or* `[]`	Empty string (ε)	`' -> 0`	See above.
`.#.`	left or right edge (beginning or end) of the string	`.#. Cons`	A word starting with a consonant (if variable `Cons` defined as such).
`|`	Union	`{th} | d`	`th` or `d` (i.e., both).
`( )`	Optionality	`(a)`	either `a` or the empty string (nothing). Equivalent to `[a | 0]`.
`\`	Term complement language	`\"+LEN"`	Not (anything except for) `+LEN` (lexical-level tag used for lenition).
`+`	One or more of the symbol preceding	`frontVow+`	One or more front vowel (if variable `frontVow` defined as such).
`*`	Zero or more of the symbol preceding	`nonPalCons*`	Any amount (including zero) of (consecutive) non-palatal consonants (if variable `nonPalCons` defined as such).
`?`	Any (single) symbol	`?*`	Zero or more of any symbol.
`:` *or* `.x.`	Crossproduct	`"+EMPH" "+3P" "+SG" "+FEM" : {-si}`	Pair an upper-level string with a lower-level string, i.e., create a relation (transducer).
`.o.`	Composition	`A:B .o. B:C`	A relation `A:B` composed with `B:C` results in the relation `A:C`.
`$`	Containment	`$["+3P" .o. L]`	All strings in language/relation `L` which (optionally) contain any prefix or any affix to `+3P`. In other words, all third person forms in `L`.
`~`	Complement language	`~[ $["+IMP" .o.  L] ]`	The language `L` minus every string (or relation) with `+IMP` (= the language without imperative forms).
`-`	Subtraction	`A - B`	The language of all strings in `A` that are not members of `B`. Equivalent to `A & ~ B`.
`&`	Intersection (or Conjunction)	`[ $[{no} "+CONJ_PART"] & ~$["+REL"] & ~$["+PRON"] ] .o. L`	All strings in language (relation) `L` that have a conjunct particle *no*, (*and*) are not relative *and* do not have an (infixed) pronoun.
`.i`	The inverse of a relation	`[ ["+" {ro} "+AUG"] : {ro} ].i`	Switches the upper and lower level around, e.g., $\frac{\texttt{r o}}{\texttt{+ r o +AUG}}$.
`.l` *or* `.u`	Projection	`@"foo.fst".l`	Selects the (in this case) lower level from a regular relation. If `foo.fst` represents the relation `<ab:cd>`, it projects the FSA `<cd>`.

Concatenation has no explicit operator; spaces between symbols result in those symbols being concatenated. Alternatively, curly brackets can be used if encoding symbols contiguously is preferred; the braces tell the compiler that the string needs to be 'exploded' into individual concatenated characters. A string such as `+VROOT` denotes one symbol in my implementation, just like, e.g, the symbol `a`. In order to avoid interpretation of '`+`' as the mathematical plus operator, it needs to be escaped. If a string like `+VROOT` is regarded as one (multicharacter) symbol (which it is in the current implementation), it must be either encoded as `%+VROOT` or as `"+VROOT"`.[10] In the `lexc` format, 'escaping' multicharacter symbols is not necessary (actually illegal). For correct string segmentation by the compiler they need to be declared in advance; if not, each of the characters will be interpreted as a one-character symbol.

Symbols and regular expression operators are invoked by the command `regex` in `foma`. Compiling the finite-state network corresponding to `a+` in `foma` is done by the procedure in Code Example 4.3 (the obligatory semi-colon signals the end of a `regex`). One is informed that the network has 2 states and 2 arcs and is cyclic (contains a loop). The `regex` command can be defined as a variable (here `myNet`) which can be used in subsequent regular expressions. The interaction with the stack is somewhat different with both operations, but lead to the same network, as the compiler output shows. The output of command `print net` includes a textual description of the states and accompanying symbols: the network's alphabet (`sigma`) contains one symbol, that is, `a`, and there is an arc with the symbol `a` from state `0` to (final) state `1`. From state `1` there is an arc with `a` going back to finite state `1`. This is the very same information as in Figure 4.1.

In the rest of this chapter, when I show replace rules in code examples, I do not include `regex` or '`;`'. It should also be noted that instead of keying in these commands in a command-line fashion, it is better practice and indeed much more convenient to use a script which contains one or more rules, which can be loaded into `foma` (with the `source` command). My convention is to use `.script` for a script file, and `.rule` for a rule. In the final part of Code Example 4.3, I illustrate this by loading in `myNet.script`, which contains a `source` command to invoke `a.rule` containing `regex a+ ;`. I generally employ the command `define` in a script to save the net as a variable for usage in a subsequent script.

[10] If + constitutes a symbol on its own, it can be encoded either as `%+` or `"+"`.

Code Example 4.3 – Illustrating various ways of compiling a regular expression in `foma`.

```
foma[0]: regex a+ ;
245 bytes. 2 states, 2 arcs, Cyclic.
foma[1]: clear stack
foma[0]: define myNet a+ ;
defined myNet: 245 bytes. 2 states, 2 arcs, Cyclic.
foma[0]: push myNet
foma[1]: print net
Sigma: a
Size: 1.
Net: 94E0DB
Flags: deterministic pruned minimized epsilon_free
Arity: 1
Ss0:    a -> fs1.
fs1:    a -> fs1.
foma[1]: clear stack
foma[0]: source myNet.script
Opening file 'myNet.script'.
Opening file 'a.rule'.
245 bytes. 2 states, 2 arcs, Cyclic.
foma[1]:
```

4.3.4 Replace rules, composition and alphabet

The tool `foma` (and `xfst`) includes templates for (variations on) the conditional replace rule format—echoing the traditional phonological rewrite rule—, which are interpreted by the compiler as a complex regular expression. An example is given in Code Example 4.4; the rule can be read read as ‘substitute the character `a` by `b` if it occurs between `x` and `y`’.[11] Multiple rewrite operators as well as multiple contexts are allowed in one rule, with also either a left or right context, or no context at all. A replace rule compiles into a transducer.

Code Example 4.4 – Template for a replace rule in `foma` (or `xfst`); compiles into a transducer.

```
a -> b || x _ y
```

Rule-based insertion of a symbol must be defined in terms of the empty string. An infinite amount of empty symbols exist between each symbol; the replace rule `0 -> a || x _ y` therefore inserts an infinite amount of `a`’s between `x` and `y`, i.e., `xay`, `xaay`, `xaaay`, etc. This is not desirable and often not intended. If the insertion of just one `a` is meant between `x` and `y`, the

[11] Note that ‘replacement’ does not, strictly speaking, involve an algorithm changing one string into another; replace rules translate into a pairing of two characters or strings (Beesley & Karttunen 2003: 133). This important difference is reflected in the observation by Antworth (1991) (section 4.2), who points out that the finite-state paradigm deals with static correspondences between strings, rather than generative rules that create intermediate forms that have no access to either the initial situation or the surface outcome.

rule format should be changed thus: `[..]  -> a || x _ y`. Replace rules target the surface level of the FST, creating new mappings as part of a subsequent transducer. The mechanics will be exemplified in section 4.6.3.

In defining replace rules, it is often convenient to encode certain underlying symbols as intermediate abstract entities or strings—perhaps reflecting an underlying morpheme or phoneme. This facilitates singling out these symbols or strings in subsequent replace rules. They often stand for an underspecified symbol of some sort whose surface forms depend on one or more intermediate rules. Various symbols of this nature are used and exemplified in this chapter, e.g., the encoding of a stem vowel as `ā` or `ī`, the future-stem consonant as `^F`, and unlenited *m* and *n* as `^M` and `^N`.[12] Cf. `alphabet.script`, section C.3.4 on page 210, for the full alphabet with variables encompassing consonants and vowels; variables aid the formulation of replace rules targeting strings containing these (semi-)surface symbols. At the moment not every variable defined is employed in the replace rules. Moreover, the alphabet is currently a mixture of abstract symbols and 'concrete' orthographical symbols. The current implementation phase has revealed that there might be some future benefit in operating with an alphabet that entirely consists of semi-surface/abstract symbols, such as phonemes (cf. section 4.6.4.3). One could imagine devising a final transducer that systematically converts (maps) these symbols into their accompanying attested Old Irish graphematic variants.

4.3.5 Morphotactic restrictions: filters and flags

The encoding of lexicons in `lexc` format—especially when they get bigger—generally leads to overgeneration of some kind: string concatenations that result in morphotactically illegal (ungrammatical) forms. This is especially true if there are separated dependencies in a language's morphology—the 'co-occurrence of morphemes that are not contiguous in the word' (Beesley & Karttunen 2003: 247–248). Instead of painfully trying to define multiple idiosyncratic 'concatenation routes' and continuation classes (section 4.3.2) that work for just a limited amount of words, it is often better to simplify the concatenation architecture and initially allow for overgeneration. This overgeneration can subsequently be restricted by using two methods, both of which are employed in my modelling of the Old Irish verbal complex.

1. *Upper-level filters* (Beesley & Karttunen 2003: 249–254): a filter specifies a lexical-level tag combination that is morphotactically invalid. This combination is subsequently deleted from the network. These tag filters apply to an overgenerating transducer, i.e. after it has been compiled (and as such are different from flag diacritics, cf. below). The general rule format for an upper-level filter incorporates the containment operator `$` and the complement language operator ∼. An example is given in Table 4.1 under the complement operator ∼.

[12] The prefixing of în conjunction with upper-case letters for surface-level symbols is a convention adopted from Beesley & Karttunen (2003).

2. *Flag diacritics* (Beesley & Karttunen 2003: Chapter 7): symbols that can be inserted alongside morphemes in the concatenation architecture to control which paths are allowed and which should be blocked in the network. Flag diacritics do not interfere with the process of inputting (analysing) or outputting (generating) a string, and can be made invisible in the output. Furthermore, they may be deleted from the network, removing illegal paths but leaving legal paths intact.

Both flag diacritics and upper-level filters are employed in the present work. The significance of flag diacritics can be illustrated by returning to Code Example 4.2. With a morpheme in `LEXICON X` as a starting point, it is not possible to specify a restricted path after leaving `LEXICON Y`, e.g., to specify that only a restricted morpheme in `LEXICON Z` should follow, or, that `LEXICON Z` should by by-passed entirely. In other words, is is not possible to 'look beyond' the subsequent lexicon. More often than not, Old Irish verb stems are found towards the middle of the verbal complex, i.e. occurring halfway in the concatenation architecture, potentially preceded by infixes in the proclitic string. Since the combination of preverb(s) and verb root is essentially arbitrary (a preverb does not go with any verb root, and simple verbs cannot be preceded by a preverb), there is only a limited number of legal paths from the start point `LEXICON A`) and the verb root. However, due to the fact that infixes 'break up' the sequence of otherwise consecutive constituents such as preverb and verb root, specifying restricted continuation classes from one lexicon to the next is no longer possible. This reflects the problem of long-distance or separated dependencies mentioned above, and these can be encoded by flag diacritics, a multicharacter symbol spelled with `@`.

The format for a flag diacritic is `@operator.feature.value@` or `@operator.feature@`. The full range is described in Beesley & Karttunen (2003: 353–356); I restrict myself to those used in the current implementation. A flag with the operator `P` sets or resets a feature to a certain value. For example, the flag `P.PV.TO` accompanies the pretonic allomorph *do* of underlying *to* to signify that a preverb has been 'seen' (`feature PV`) which is classified as `TO` (its `value`). Setting this `feature` to the indicated `value` allows us to either disallow (with operator `D`) or, conversely, require (with operator `R`) the `feature` associated with this morpheme. For example, there is a dependency relation between *to* and *lēc* in the compound verb *do·léici*, for which one can employ a combination of `P.PV.TO` and—accompanying the potentially non-contiguous verb root—`R.PV.TO`. Simple verbs by definition are not preceded by a lexical preverb. This class as a whole can be accompanied by `D.PV`, causing all 'concatenation routes' originating from any preverb to be blocked. More examples with flag diacritics will follow in this chapter.[13]

The alternative, in this case less desirable, option is to use upper-level filters for illegal co-occurrences of especially preverb and verb root tags after compilation of an overgenerating lexicon. These constraints are not straightforward to apply post-hoc due to the essentially idiosyncratic combination patterns of preverbs and verb roots (as mentioned above and in section 4.7). Furthermore, flag diacritics have a restraining effect only at runtime, keeping the

[13] For the full range of flags for preverbs cf. `LEXICON Preverb` in `proclitic.lexc` (section C.1.2 on page 179). Their matching counterparts can be seen in the stem lists in the Stem entry files (`.txt`) (section C.4 on page 222).

transducer small, while upper-level filters apply after an overgenerating and overrecognising network has been compiled; as such, the latter operation substantially increases the amount of states in a network (Beesley & Karttunen 2003: 297–300). This, however, was not the main reason for partly employing flag diacritics in my framework, as the difference in processing time compared to upper-level filters was found to be negligible.

Basic dependency relations with proclitics and verb stems (simple or compound, deurotonic or prototonic) can be conveniently and and transparently constrained with flag diacritics as will be demonstrated throughout this chapter, and especially in section 4.7.2. Other dependencies involve more complex restriction specifications for which the post-hoc upper-level filters were found to be more suited (cf. section 4.8). One example with rather intricate dependencies concerns the verb form (*hóre*) *nondob·molor-sa* 'because I praise ye', already discussed in section 2.2.6, reintroduced here as (20) and further discussed in section 4.4.2.

(20) **((h)óre) no-ndob-mol-or=sa**
((h)óre) PART-INFIX-mol-SUFF=SUFF
(because) CONJ_PART-REL\PRON.2PL-praise-PRS.IND.1SG=EMPH.1SG
'(because) I praise ye' (non-relative -dob-)

4.3.6 Regression testing

Testing with stem entries and inflectional rules was carried out using an incremental approach called *regression testing* (Beesley & Karttunen 2003: 334–335). This method is based on subtraction (cf. Table 4.1) of networks. Typically before adding a stem entry or encoding a replace rule, the network is compiled and saved as a binary file, say, network `A`. The network is subsequently recompiled with the additional stem entry or rule as, say, network `B`. The result of subtracting the 'new' network from the 'old' one (`A - B`) equals every string (word) in network `A` that is not in network `B`; this is effectively the set of strings (words) that were lost after making changes to the concatenation infrastructure and/or replace rule(s). Conversely, subtracting the 'old' network from the 'new' one (`B - A`) gives us those strings that were added to the network. One obviously wants to make sure that lost strings reflect ungrammatical or undesired words, and that added ones are grammatical or desired. Scripts for these operations are found in section C.3 on pages 209–212.

Regression testing has proved to be invaluable for making sure that the FST contains morphotactically legal lexical-level strings as well as correctly formed orthographical surface forms. The five main verbs used during implementation and preliminary testing are W1 *marbaid* 'kills' (Appendix B, pages 173–174), W1 *ad·ella* 'approaches, visits, touches', W2a *brissid* 'breaks', W2a *léicid* 'lets, leaves' (Appendix B, page 172) and W2a *do·léici* 'lets go, releases, casts', etc. The system implemented based on these five verbs has been tested against inflected forms of 27 verb lemmas from the text *Táin Bó Fraích* (cf. Table 5.2, page 111).

4.4 Modelling the verbal complex: choices and challenges

4.4.1 Lexical and surface level description

The lexical or upper level in the transducer for Old Irish verbs consists of underlying forms, which, in the case of verbs, means the (abstract) root shape. This was motivated by three insights:

1. The usual citation form, the independent prs. ind. 3sg., i.e., the deuterotonic form in the case of compound verbs (in eDIL and other dictionaries/vocabularies), often does not transparently show the underlying forms that one has to operate with to explain the inflectional forms across paradigms.

2. Encoding underlying/root forms in the lexical/upper level ensures that all surface forms can be easily generated on the basis of unambiguous tags.

3. Underlying forms also show the diachronic development (pre-forms) that are often insightful in terms of explaining the surface form, allowing for interoperability with computational implementations of other historical Indo-European languages, or, indeed, Proto-Indo-European.[14]

For lexical/upper-level tags, their grammatical description and references to Chapter 2, cf. the Glossary on page xix.

4.4.2 Typographical variability and morphotactic dependencies

When talking about the verb in Old Irish one really refers to the verbal complex: the combination of (1) proclitics (pretonic prefixes, e.g., preverbs, augments and conjunct particles), (2) the verb stem, and (3) the endings. I have aimed for correct analysis and generation of this morphologically highly synthetic 'word'. Two complementary implementation challenges arise relative to the verbal complex: (1) spacing and (2) absence of spacing. This variability translates into morphotactic dependencies (including separated dependencies, cf. section 4.3.5) either string-internally or across strings separated by space. If a space occurs between he proclitic string and the subsequent string with the verb root, the same morphotactics obviously apply, but across separated strings that together constitute the verbal complex. In other words, although the FST implementation is initially restricted to words in isolation, many grammatical analyses must be anticipated to allow for subsequent morphotactic disambiguation across word boundaries (not part of the work described here).

These constraints, when implemented, target inflectional non-possibilities, but also involve word class disambiguation (POS tagging). For example, in order to facilitate correct morphosyntactic tagging of the sequence *ná fer!* in a text, one wants to be able to successfully

[14] See for example *Proto-Indo-European Lexicon* at `http://pielexicon.hum.helsinki.fi/`, a generative etymological dictionary of Indo-European languages, which is also implemented using the finite-state toolkit `foma`.

generate all potential grammatical analyses of *fer*; this includes the noun reading of *fer*, i.e., ‘man’, or, alternatively, one of a few possible inflected forms in the paradigm of the verb *feraid* ‘supplies’, etc.[15] (the imp. 2sg. in this case). The current implementation also caters for the separation of the first preverb in a deuterotonic compound verb. Hence, in the context of isolated word parsing, *fer* might be the ‘deuterotonic part’ of a compound such as *fo·fera* ‘brings about’ etc., for example, prt. 3sg. *fo␣fer*. All possible verb readings (as well as the noun reading) must be generated regardless of the fact that the element *fer* is not immediately consecutive to the preceding preverb, augment or conjunct particle, even though its inflection is often dependent on the latter’s presence.[16]

Conversely, if the goal is to facilitate recognition of a consecutive string and to generate all morphotactically valid possibilities, a morphological parser should also be able to deal with cases such as *ná·fer*, and, ideally, with alternative typographical practices such as *ná-fer* and *náfer*. Editors employ different strategies to mark the proclitic juncture in the verbal complex, although usage of the mid-high dot ‘·’ has emerged as the typographical standard in text editions and grammars. An edition with the transcribed text being verbatim to the manuscript[17] obviously does not contain these word/morpheme segmentation markers. Catering for cases such as *nondobmolorsa* (Würzburg glosses, 14c18),[18] repeated here as (21), is not trivial; a significant amount of morphotactic restriction rules, including separated dependencies, would need to be encoded for this form, as shown below.

(21) **((h)óre) no-ndob-mol-or=sa**
((h)óre) PART-INFIX-mol-SUFF=SUFF
(because) CONJ_PART-REL\PRON.2PL-praise-PRS.IND.1SG=EMPH.1SG
‘(because) I praise ye’ (non-relative -dob-)

- The ‘empty’ particle *no* can only precede a simple verb.
- *no* demands the conjunct ending set.
- *no* is obligatory with:
    1. the so-called secondary endings (imperfect, past subjunctive, conditional), or,
    2. other tense/mood combinations but only when its function is to support an infixed pronoun
        - excluding relative forms, where *no* is often obligatory with simple verbs,
            - but not where a person/number combination has a special absolute relative ending (3rd persons and 1pl.)
                - provided that there is no need to support an infixed pronoun.

[15] cf. `http://dil.ie/21676`.

[16] Note that the term ‘dependent’ is used in a broad sense here; in Old Irish grammar, the meaning of this term is restricted to verbs that are preceded by a conjunct particle (cf. section 2.2.1).

[17] A text transcribed by the editor verbatim to the manuscript is known as a diplomatic edition.

[18] It must be noted that the verb *molaid(ir)* ‘praises’, which in Old Irish is a deponent verb, is not part of the set of verbs focused on in this thesis.

- The emphasising particle -*sa* must agree in person and number with either the verb ending or the infixed pronoun (the form -*sa* can only agree with the verb ending here).

Admittedly, the morphemes constituting (21) are transparent and unambiguous; there is no other grammatical interpretation possible (for example, -*dob*- can only be 2pl.). This is, however, often not the case. Compare, for example, the various possible and impossible readings in Table 4.2. The ambiguity here is caused mainly by the non-surfacing of the infixed pronoun masc./neut. *a* when following *ní*, and the fact that the emphasising particle -*sem* (not considering the context) can be both singular masc. and neut., as well as 3pl. My aim is to allow for all possible lexical-level parses of a verb form, while restricting wrong ones as much as possible.

Table 4.2 – List of parses including ungrammatical ones (with a strike-through) for the ambiguous orthographical form *níléicisem* (*ní-léic-i=sem*).

Morph. gloss	translation (if applicable)
CONJ_PART.NEG-let-PRS.IND.3SG=EMPH.3SG.M	'*he* does not let'
CONJ_PART.NEG-let-PRS.IND.3SG=EMPH.3SG.N	'*it* does not let'
~~CONJ_PART.NEG-let-PRS.IND.3SG=EMPH.3PL~~	
~~CONJ_PART.NEG-let-PRS.IND.2SG=EMPH.3SG.M~~	
~~CONJ_PART.NEG-let-PRS.IND.2SG=EMPH.3SG.N~~	
~~CONJ_PART.NEG-let-PRS.IND.2SG=EMPH.3PL~~	
CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.3SG=EMPH.3SG.M	'*he* does not let him' or 'does not let *him*'
CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.3SG=EMPH.3SG.N	'*it* does not let him'
~~CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.3SG=EMPH.3PL~~	
CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.3SG=EMPH.3SG.M	'*he* does not let it'
CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.3SG=EMPH.3SG.N	'*it* does not let it' or 'it does not let *it*' or 'does not let *it*'
~~CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.3SG=EMPH.3PL~~	
CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.2SG=EMPH.3SG.M	'you do not let *him*'
~~CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.2SG=EMPH.3SG.N~~	
~~CONJ_PART.NEG.PRON.3SG.M-let-PRS.IND.2SG=EMPH.3PL~~	
CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.2SG=EMPH.3SG.N	'you do not let *it*'
~~CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.2SG=EMPH.3SG.M~~	
~~CONJ_PART.NEG.PRON.3SG.N-let-PRS.IND.2SG=EMPH.3PL~~	

The present work aims to also facilitate successful recognition in this challenging scenario, not in the least since advances in optical character recognition (OCR) for medieval manuscripts are expected in the near future, resulting in texts with various spacing conventions to become increasingly more available. A question not addressed in this thesis is whether morphological analysis should be preceded by tokenization, or whether both activities should be integrated as part of one model. Automatic tokenization for Old Irish has only very recently started to receive attention. Doyle, McCrae & Downey (2019) report on the development of neural machine-learning methods for tokenizing the Old Irish Würzburg glosses. It is hoped that future collaboration will generate advances in word-level parsing for Old Irish.

As said above, I aim for correct generation of a multi-morpheme string. However, one could, in theory, create any combination of proclitic element and verb root. If one were to play the devil's advocate, one could say: 'why bother trying to painfully restrict completely impossible combinations of, say, preverbs and verb roots such as *ess* and *marb*, to give **as·marba*; such a parse will never come up since a form like this does not occur in a text'. In other words, one option is to entirely focus on coverage by ignoring the morphotactics of Old Irish, resulting in a vast amount of ungrammatical strings in the transducer (overgeneration). However, as Table 4.2 illustrates, ignoring the morphotactics of a language will come at a cost: ambiguous spellings (homographs) may get various interpretations, of which generally only a limited number are actually grammatically correct. Only morphotactically valid strings should remain. As discussed in section 4.3.5, the finite-state toolkit has instruments to avoid the analysis and generation of non-possibilities.

Grammatical ambiguity often cannot be entirely resolved due to the underspecified nature of Old Irish orthography. As can be seen in Table 4.2, the infixed pron. 3sg. neut. is accompanied by lenition, but this is not marked in the spelling with *l*. The infixed pron. 3sg. masc. is accompanied by nasalisation, but [L] is not obligatorily spelled with the digraph *ll*-, i.e., one finds both *l*- and *ll*- in this case. Section 4.8.2 deals with the implementation of initial consonant mutations and related matters.

While establishing the recognition rate of orthographical surface forms is priority in this thesis (cf. Chapter 5), substantial efforts have been made to exclude morphotactically invalid strings from my transducer, to avoid the above-mentioned problem with ambiguous surface-level forms. Moreover, it is expected that generation of mostly grammatically correct surface forms will have applications in the future. For example, easy and comprehensive access to normalised forms will be beneficial to linguists who want to compare normalised and expected forms against the attested evidence, and it will greatly facilitate students of Old Irish. Restricting generation—as much as possible—to *possible* forms is also crucial for accurate and unambiguous lemmatisation and standardisation (or, rather, normalisation), for which cf. section 6.4.4.

4.4.3 A solution: two main lexicons

I have devised two lexicons for the verbal complex which are combinable: a proclitic lexicon and a lexicon that incorporates verb stems with the relevant ending sets (Figure 4.5). The rationale for operating with two separately compiled lexicons is that both elements—pretonic elements and the part that contains the verb root—may either be separated or consecutive in a text, as detailed in section 4.4.2. This means that I have pre-empted what would be covered in the tokenization/pre-processing stage in a standardised-language scenario (e.g., establishing word or morpheme boundaries by standard non-alphabetical characters including spaces and hyphens).

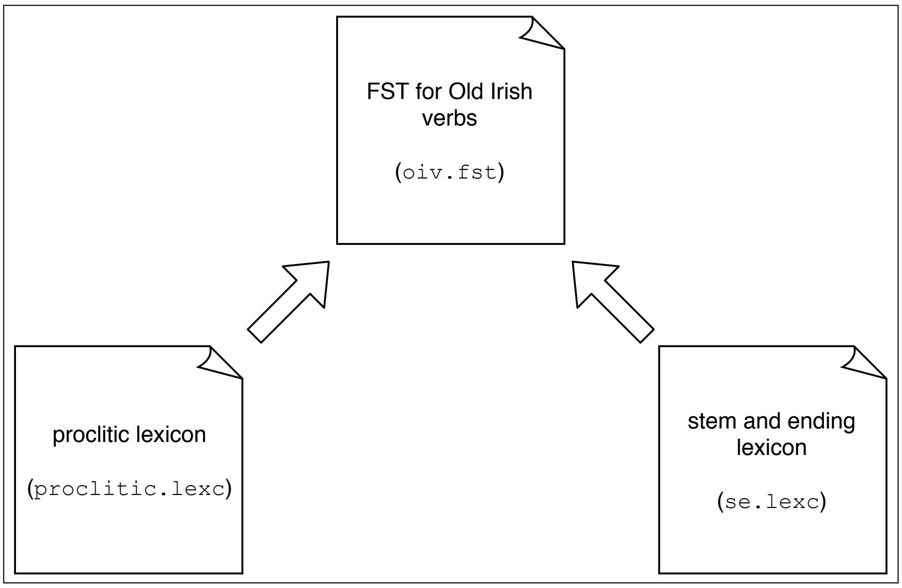


Figure 4.5 – The FST implementation of the Old Irish verbal complex: two main lexicons.

The endings include the emphasising particles or *notae augentes*, as well as the suffixed pronouns. Suffixed pronouns can only appear with absolute endings and occur almost exclusively with 3sg. forms. I encoded the latter as part of the stem-and-ending lexicon but for abs. 3sg. endings only (non-3sg. absolute verb forms with suffixed pronouns can be manually added to the lexicon on a one-to-one basis).

Both lexicons can be combined to facilitate recognition of, say, *ní* on its own, *léici* on its own, and the consecutive string *ní(·)léici*. Note that the current implementation also facilitates the separation of preverb and verb root (*do* and *léici* are individually recognised). One can thus further add to the list of potential correct parses for *léici* a string starting with the (deutero)tonic element of the compound *do·léici*, 2 or 3sg. prs. ind. conjunct. Before delving into the possibility of combining the two lexicons, I will illustrate the code for each separately.

4.5 The proclitic lexicon (`proclitic.lexc`)

Pretonic preverbs and augments, as well as conjunct particles, are all compiled into one `lexc` file `proclitic.lexc`, as they may all be followed by a designated set of infixed clitics—the object pronouns, which are also part of this lexicon file. If one created separate lexicons for the preverbs, augments and conjunct particles, one would end up having to encode duplicate infixed pronoun entries. Code Examples 4.5 and 4.6 show a fragment of `proclitic.lexc` (for the matching flag diacritics with stems cf. section 4.7.2).

Code Example 4.5 – Snippet of `proclitic.lexc` (section C.1.2 on page 179) showing the proclitic lexicons containing the preverbs, conjunct particles and augment *ro*.

```
!*** Root = start ***

LEXICON Root
                Preverb;
                conjPart;
@P.PART.NO@     No;
                Ro;
```

```
LEXICON Preverb
```

```
@P.PV.TO@           TO;
```

```
LEXICON TO
to+PV1:do           #;
to+PV1:do           pronA;
to+PV1+REL+LEN:do   #;
to+PV1+REL+NAS:do   #;
to+PV1+REL+LEN:do   pronC;
to+PV1+REL+NAS:don  pronC;

!*** Conjunct particles ***

LEXICON conjPart
ní+CONJ_PART+NEG:ní     #;
ní+CONJ_PART+NEG:ní     pronA;
ní+CONJ_PART+NEG:ní     roNonRel;
```

```
!*** Ro ***

LEXICON Ro
    roNonRel;
```

```
LEXICON roNonRel
+ro+AUG:ro  #;
+ro+AUG:ro  pronA;
```

Code Example 4.6 – Snippet of `proclitic.lexc` (section C.1.2 on page 179) showing part of the infixed pronoun lexicons.

```
LEXICON pronA
+PRON+A+1P+SG+LEN:^M            #;
+PRON+A+2P+SG+LEN:t             #;
+PRON+A+3P+SG+MASC+NAS:^PRONa   #;
+PRON+A+3P+SG+FEM:s             #;
```

```
+PRON+A+3P+SG+FEM+NAS:s          #;
+PRON+A+3P+SG+NEUT+LEN:^PRONa    #;
```

```
LEXICON pronC
+PRON+C+1P+SG+LEN:dom        #;
+PRON+C+1P+SG+LEN:dum        #;
+PRON+C+1P+SG+LEN:dam        #;
+PRON+C+1P+SG+LEN:damm       #;
+PRON+C+2P+SG+LEN:dat        #;
+PRON+C+2P+SG+LEN:dit        #;
```

Only a few replace rules accompany `proclitic.lexc`, partly reflecting the more predictable nature of pretonic (i.e., unstressed) elements. The rule sequence in Code Example 4.7 takes care of vowel coalescence with the infixed pronoun 3sg. masc./neut. after *no*, *ro* and *do* (*o* becomes *a*) and *ní* (*a* does not surface after this negative particle).[19] Another rule rewrites the underspecified symbol `^M`, for unlenited /m/, into either *m* or *mm*, e.g., *dom* and *domm*. The augment *ro* may either occur word-initially or follow a conjunct particle. However, in both cases the same `LEXICON` is used. If *ro* happens to be in initial position, it contains initial '+' on the upper level; if so, a rewrite rule caters for the deletion of this symbol.

Code Example 4.7 – A sequence of replace rules dealing with the infixed pronoun class A 3sg. masc. / neut. after the augment *ro* and conjunct particles.

```
o -> 0 || n|r|d _ "^PRONa" .o.
"^PRONa" -> 0 || {ní} _  .o.
"^PRONa" -> a
```

4.6 The stem-and-ending lexicon (`se.lexc`): from semi-surface to surface forms

4.6.1 Monolithic stems: recapitulation

In Chapter 2, especially section 2.3 and 2.4, I have illustrated that a significant amount of allomorphic variation can be seen with verb stem formation. The non-transparent allomorphic variation is essentially due to the outcome of a rigid stress system (phonology) which results in 'syntactically governed accent shifts' (Stifter 2009: 89) with compound verbs (and simplexes with the augment *ro*). That is, the very same compound verb is stressed either on its first or second element, creating entirely different-looking stem variants which are hard to relate back to the underlying root forms. This makes the formulation of a stem entry far from trivial.

The schemas introduced in Chapter 2 have made this clear. In its simplest form, abstracting from simple, compound and dependency, the verbal complex has the schematic structure

[19] This rule was updated after testing, cf. section 5.3.1.

(CONJ_PART) PV* (AUG) PV* VROOT E. If one blindly applied the morphological concatenations without regard to phonology, one would get, for example, *to-ro-lēc* (PV AUG VROOT) and *ní-to-ro-lēc* (CONJ_PART PV AUG VROOT), where the morphological derivation is quite far removed from surface or orthographical forms such as prt. 3sg. *do·reilic* and *ní·tarlaic*, respectively. Considering the morphotactic schema as reflecting a combinatorial problem within a computational framework, even when assisted by the (tentatively formulated) positional hierarchy of preverbs in McCone (1997: 90), is therefore hardly useful: the combination of preverbs and verb root—assuming, first of all, that an exhaustive set of existing combinations is available—generally does not get us very far in terms of the surface form.

Although the finite-state morphology paradigm can in principle be employed for both concatenative (affixation) and non-concatenative morphology (stem-internal processes such as ablaut), the computational paradigm will not inform the linguist what the most logical stem entry is. He or she needs needs to manually define a string that facilitates trivial computational modelling of morphological rules (but cf. section 6.4.5 for a morphological guesser). In section 2.4.1 I have argued for the employment of a so-called *monolithic stem* for the purpose of computationally modelling the Old Irish verbal system: a unit taken as-is, not derived from individual morphemes by computational rule application. Identifying a stem entry that facilitates straightforward suffixation of endings in a verb's paradigm has been one of the main challenges as well as an important outcome of this project. I will now illustrate how monolithic stems are reflected in the actual code.

4.6.2 Encoding the monolithic stem

The goal is to efficiently code a stem that allows for simple morphological rules and easy modelling of ending variation. Ending variation (including stem consonant allomorphy/orthographical variation) is often the result of syncope, which will be illustrated in section 4.6.5 with the fut. 1pl. examples *ad·ellfam*, *ad·eillfem* vs. *·aidlibem* 'we will approach, visit, touch'. It was found that the stem should be encoded in a semi-surface, pre-syncope form. This insight has led to an approach whereby surface forms are derived in a stepwise fashion by syncope rules. Thus, apart from defining one (or more) monolithic stems for each verb, which are very much surface forms, these stems should at the same time incorporate vowels that are under certain circumstances deleted.

The insight that the base should be in a pre-syncope format agrees with another strategy in the implementation, namely, the encoding of weak verb stems including a stem vowel.[20] The weak verb types W1 and W2 (cf. section 2.2.7) are alternatively referred to as *a*- and *i*-verbs by Thurneysen (1946: §§ 521–525, 546), who otherwise uses the classification AI and AII, respectively. I am invariably using macrons (`ā` and `ī`), but without implying that these always represent historical stem vowels. It proved to be efficient and insightful to define a stem such as `marbā` for the verb *marbaid* 'kills' and `lēicī` (including augmented `reiləcī`) for the

[20] I am thankful to Prof. David Stifter, Professor of Old Irish at Maynooth University, for bringing this insight to my attention.

verb *léicid* ‘lets’. Code Example 4.8 and Code Example 4.9 show the stem entries while the continuation lexicons in Code Example 4.10 add the relevant W1 and W2a stem vowel.[21]

Code Example 4.8 – Stem entry for the simplex *marbaid*, part of simpleW1.txt (section C.4.3 on page 223).

```
+marb+VROOT:marb            W1;
```

Code Example 4.9 – Stem entries for the simplex *léicid*, part of simpleW2a.txt (section C.4.4 on page 224).

```
+lēc+VROOT:léic             W2a;
+ro+AUG+lēc+VROOT:reiləc    W2a;
```

Code Example 4.10 – Continuation lexicons for W1 and W2a stem vowels as part of se.lexc (section C.1.3 on page 184).

```
LEXICON W1
+W1:ā   weakStemFormation;

LEXICON W2a
@P.W2a.ON@+W2a:ī    weakStemFormation;
```

4.6.3 Stem consonants, endings and suffixes

The se.lexc file contains the stems and endings. The start of the continuation classes is shown in Code Example 4.11. In the first lexicon, the binary distinction between simple and compound verbs is defined. While this work focuses on weak verbs, the substantive verb (substV) as well as the copula, due to their high frequency in texts, are included. The copula is always unstressed and would lead to too many nonsense combinations when integrated into the main verb lexicon. It is was therefore decided to create a full-form lexicon for the copula (section C.1.1 on page 175). However, the equally irregular substantive verb behaves somewhat more like a ‘normal’ verb in that it can occur in stressed position following the proclitic boundary, and has therefore been integrated into the main verb lexicon se.lexc. Strong verbs are not dealt with in this work, but their inclusion in the lexc concatenation infrastructure has been anticipated, as can be seen in the comments (preceded by !) in Code Example 4.11.

[21] The same present indicative ending set is used for W1 and W2a; the latter’s accompanying flag makes sure that prs. ind. 1sg. *-u* only gets suffixed to W2a verbs (Code Example 4.12).

Code Example 4.11 – Start of continuation classes in `se.lexc` (section C.1.3 on page 184).

```
!\\\\\ BEGIN CONTINUATION CLASSES /////

!\\\\\ STEMS /////

!*** Root = start ***

LEXICON Root
@D.PV@          simpleStems;
@D.PART.NO@     compoundStems;
                substV;

!*** simple vs. compound ***

LEXICON simpleStems
    simpleW1;
    simpleW2a;
! continuation classes for strong types can be added later

LEXICON compoundStems
    compoundW1;
    compoundW2a;
! continuation classes for strong types can be added later

!*** Shell script inserts stems, maintained in separate files, for <PLACEHOLDERS> ***

LEXICON simpleW1
<INSERT SIMPLE W1 STEMS>

LEXICON simpleW2a
<INSERT SIMPLE W2a STEMS>

LEXICON compoundW1
<INSERT COMPOUND W1 STEMS>

LEXICON compoundW2a
<INSERT COMPOUND W2a STEMS>

!*** Weak stem formation ***

LEXICON W1
+W1:ā   weakStemFormation;

LEXICON W2a
@P.W2a.ON@+W2a:ī    weakStemFormation;

```

```
LEXICON weakStemFormation
+PRS+IND:0       weakPresIndEndings;
+IMP:0           weakImpEndings;
+IPF:0           secEndings;
+PRS+SUBJ:0      aEndings;
+PAST+SUBJ:0     secEndings;
+FUT:^F          aEndings;
+COND:^F         secEndings;
+PRT:^S          sPretEndings;
+PRT+PASS:θ      pretPassEndings;
```

Simple verbs (stems) cannot be preceded by a preverb, hence the flag diacritic with the D (disallow) feature to make sure that under no circumstances is a lexical preverb prefixed. Compound verb forms, in either deuterotonic or prototonic form, cannot under any condition be preceded by *no*; hence the second flag diacritic @D.PART.NO@ here. One ultimately arrives at (monolithic) stem entries for each verb (e.g., Code Example 4.8 and Code Example 4.9), which are part of separate stem entry lists (section C.4 on page 222) and inserted here by unix commands as part of a shell script (section C.5 on page 224).

The present indicative endings are shown in Code Example 4.12. The 'semi-surface' stems and endings are subsequently subject to a list of surface-level rewrite rules encoded in se_1_bare.script (section C.3.11 on page 216), based on consonant and vowel variables defined in alphabet.script, for which cf. section C.3.4 on page 210. For example, ^F represents the future stem consonant and is a lower-level trigger to facilitate replace rules for this tense later on. As can be seen from Code Example 4.11, however, the present indicative, imperative, imperfective and present subjunctive inflection do not actually result in a (tense/mood) stem rewriting process: although the relevant tags are added on the upper, lexical level, the lower level contains the empty string, encoded as 0 (zero). This encoding reflects the justification for the focus on weak verbs: those tense/mood stems that do involve an additional stem formation process are composed by (predictable) suffixation only.

The concatenation architecture illustrated here results in, for example, prs. ind. 3pl. absolute (surface-level) marbāt' '(they) kill'. Figure 4.6 and Figure 4.7 illustrate how subsequent replace rules lead to the correct surface form. For each individual intermediate stage (box in Figure 4.6) there is an upper and lower form. By compressing the whole stack of intermediate stages through composition (.o.), one ends up with just one transducer (box) with the final upper level and lower level. The initial upper-level string generally remains the same, inherited from the lexc file, while the lower-level form is manipulated by rules. However, one may want to create rules to target the upper-level tags as well, if desired. The finite-state paradigm allows one to go from lexical level to surface form and back again, i.e., it is bidirectional. If used in an upward direction the transducer is used in recognition mode (producing lemma and tags = morphological analysis), while a downward direction represents the generation mode (usually producing orthographical strings).

Code Example 4.12 – The weak present indicative endings as part of `se.lexc` (section C.1.3 on page 184).

```
!\\\\\ ENDINGS /////

LEXICON weakPresIndEndings
+ABS+1P+SG:^M'              Emph1sg;
@R.W2a.ON@+ABS+1P+SG:u      Emph1sg;
+ABS+2P+SG:i                Emph2sg;
+ABS+3P+SG:θ'               suffAbs3sg;
+ABS+3P+SG+REL:s            Emph3sg;
+ABS+1P+PL:^M'i             Emph1pl;
+ABS+1P+PL+REL:^M'e         Emph1pl;
+ABS+2P+PL:θ'e              Emph2pl;
+ABS+3P+PL:t'               Emph3pl;
+ABS+3P+PL+REL:^V^D'e       Emph3pl;
+CONJ+1P+SG:^M'             Emph;
@R.W2a.ON@+CONJ+1P+SG:u     Emph;
+CONJ+2P+SG:i               Emph;
+CONJ+3P+SG:0               Emph;
+CONJ+1P+PL:µ               Emph;
+CONJ+2P+PL:θ'              Emph;
+CONJ+3P+PL:t               Emph;
+PASS:0                     pass1Endings;
```

```
marb+VROOT+W1+PRS+IND+ABS+3P+PL
-------------------------------
            marbāt'
```

`8_se_phon_stem_vow.rule` (section C.2.8 on page 201).

```
[ [..] -> i || ā _ palCons ] .o.
[ā -> a ] .o.
```

```
marb+VROOT+W1+PRS+IND+ABS+3P+PL
-------------------------------
           marbait'
```

`12_se_del_pal_markers.rule` (section C.2.12 on page 203).

```
regex [' -> 0 ] ;
```

```
marb+VROOT+W1+PRS+IND+ABS+3P+PL
-------------------------------
            marbait
```

Figure 4.6 – A schematic illustration of intermediate levels during rule application, using an example verb form.

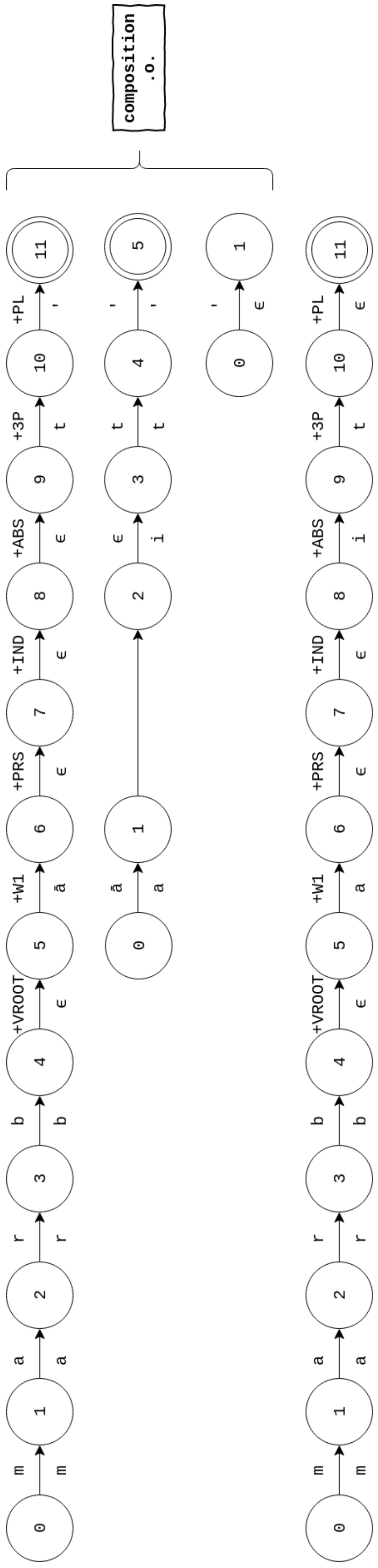

composition
.o.
0 m:m 1 a:a 2 r:r 3 b:b 4 +VROOT:ε 5 +W1:â 6 +PRS:ε 7 +IND:ε 8 +ABS:ε 9 +3P:t 10 +PL:- 11
0 â:a 1 2 ε:i 3 t:t 4 -:- 5
0 -:ε 1
0 m:m 1 a:a 2 r:r 3 b:b 4 +VROOT:ε 5 +W1:a 6 +PRS:ε 7 +IND:ε 8 +ABS:i 9 +3P:t 10 +PL:ε 11

Figure 4.7 – An FST network representation of intermediate levels during rule application, using an example verb form.

By incorporating the stem vowel in the base, the endings can be defined in an abstract way. Most grammar books would have the prs. ind. (and subj.) 3sg. abs. ending as *-aid* or *-id* (e.g., Stifter 2009: 91–92), i.e., endings need to be individually specified for the types W1 and W2a. Encoding this ending as `θ` instead also facilitates the suffixation of suffixed pronouns, where the inflectional ending invariably surfaces as *th* (e.g, *ber-th-us* 'carries her/them', cf. example (8), Chapter 2). Table 4.3 contrasts some 3sg. endings in Stifter (2009) with their encoding in the present implementation, including rewrite rules to arrive at the latter.[22]

Table 4.3 – Contrasting some third-person singular endings in Stifter (2009) and the encoding in the current implementation.

	Stifter (2009)	Current implementation	Replace rules	Examples

prs. ind./subj. 3sg. abs.	*-aid*/*-id* (Present Ia/IIa/*a*-endings)	`θ'`	`[ [..]  -> i || ā _ palCons]`
			`.o. [ā -> a] .o. [ī -> i] .o.`
			`[θ -> {th}|d]`
				`marbāθ'` → `marbaith/-d`
				`léicīθ'` → `léicith/-d`

prs. ind. 3sg. conj.	*-a*/*-i* (Present Ia/IIa)	`0`	`[ā -> a] .o. [ī -> i]`	`marbā` → `marba`
				`léicī` → `léici`

prs. subj. 3sg. conj.	*a* (*a*-endings)	`a`	`[ ī -> e || _ [nonPalCons+`
			`[ Vow|.#.|"-"]] | a] .o.`
			`[ā -> 0 || _ a|u|∅]`
				`marbāa` → `marba`
				`léicīa` → `léicea`

prt. 3sg. abs.	-s^j (s I)	`'`	`[ [..]  -> i || ā _ palCons]`
			`.o. [ā -> a] .o. [ī -> i] .o.`
			`[' -> 0] .o. [^S -> s]`
				`marbā^S'` → `marbais`
				`léicī^S'` → `léicis`

prt. 3sg. conj.	*-∅* (s I)	`∅`	`[^S -> 0 || _ ∅] .o.`
			`[ā -> 0 || _ a|u|∅] .o.`
			`[ī -> 0 || _ e|i|∅] .o.`
			`[∅ -> 0]`
				`marbā^S∅` → `marb`
				`léicī^S∅` → `léic`

One ending set, accompanied by subsequent rules, can be employed to deal with the variation in endings with W1 and W2a verbs. The prt. 3sg. conj. ending `∅` communicates that the stem vowel and the stem consonant `^S` need to be deleted, to arrive at *·marb* and *·léic*, respectively.[23] In the present indicative paradigm one encounters a `0` for 3sg. conj., which is different from `∅`: the `0`-ending translates into an endingless suffix with only the stem vowel (or what remains from it): *·marba* and *·léici*, respectively. The ending `0` is therefore different from Stifter (2009), who works with stems like *marb* and *léic*, where *-∅* (rather than `0`) means a null-ending. In his system, the rather divergent endings *a* and *i* in the prs. ind. 3sg. conj. need

[22] The rules employed are `3_se_phon_lowering.rule` (section C.2.3 on page 198), `5_se_phon_del_stem_vow.rule` (section C.2.5 on page 199), `8_se_phon_stem_vow.rule` (section C.2.8 on page 201), `12_se_del_pal_markers.rule` (section C.2.12 on page 203) and `13_se_phon_orth_cons.rule` (section C.2.13 on page 203). Note that there are many more replace rules than the ones given, so the rules do not necessarily occur contiguously as presented here.

[23] The same ending encoding is used for the imp. 2sg. and subj. 1sg. conj., resulting in deletion of the stem vowel only (there is no stem consonant with the so-called *ā*-subjunctive, utilised by most weak verbs).

to be specified for both stem classification types, whereas in my implementation a null-ending (`0`, which in the regular expression language means the empty string) and very straightforward replace rules suffice for this ending. The symbol `θ` can be used to either rewrite to *th* or *d*, reflecting (diachronic) spelling variation in Old Irish.

The ‘extra’ vowel encoded as part of the stem also simplifies a rule-based approach to syncope. For example, in the *s*-preterite, one encounters forms like 3pl. abs. *léic*†*sit*, *marb*†*sait* and conj. *·léic*†*set* and *marb*†*sat*, where, again, a divergence in ending can be observed. Moreover, when preceded by stressed *ro*, *·léic*†*set* becomes *·reil*†*ciset*,[24] with an *i* appearing before the stem consonant *s*. The ending format contrasts with the system in Stifter (2009), who gives ‘underspecified’ 3pl. abs. *-s(a)it* and conj. *-sat*, *-set*. Stifter (2009: 92) accounts for the potential ending variation by noting:

> The *s* is that of the stem. The main difference, the presence (or not) of a vowel between the *s* and the ending, is just an automatic consequence of divergent syncope patterns, just like the difference in palatization of the *s*.

This conditionally determined ending variation can be easily encoded with single ending sets when using `ā` and `ī` as part of the stem, which is syncopated if in an even-numbered syllable. The implementation of syncope and concomitant changes to consonant quality is described in section 4.6.4. Section 4.6.5 contains a fully worked-out example with the *f*-future stem consonant using the compound verb *ad·ella* ‘approaches’, etc.

Palatalisation markers (') are added to deal with the orthography of unstressed vowels. For example, while prs. ind./subj. 3sg. abs. `lēicīθ'` is close to the orthographical surface form, in `marbāθ'` an *i* needs to be inserted before `θ'`, which is catered for by the rule `[..] -> i || ā _ palCons`. The inflectional endings in Table 4.3 are the result of assimilation with the stem vowel (`ā`/`ī`). When a stem consonant is present, as in the case of the preterite, ending vowels in closed syllables are encoded as `ə`. The 3pl. preterite endings discussed above, for example, are produced by a stem such as `marbā`, the stem consonant `^S` and an ending `ət'` or `ət` for absolute and conjunct, respectively. The final ‘value’ of schwa depends on the quality of the preceding consonant (the quality of the final consonant is ‘known’, as it is encoded as part of the ending). The quality of the preceding consonant is subject to variation due to syncope, so the rewriting of `ə` must follow the outcome of consonant quality assimilation (section 4.6.4.3). When consonant quality has been established, the rules pertaining to `ə` are as presented in Code Example 4.13, emulating Table 2.3 in Chapter 2.

Palatalisation markers are ‘cleaned up’ after consonant quality assimilation has been applied, and after vowels have been added and rewritten based on these markers. Palatalisation markers at the end of a string are also employed in replace rules that establish the surface forms of the 1sg., 2sg. and 3p. emphasing particles[25] when immediately preceded by a consonant (otherwise the quality of the final vowel in the ending is important). Emphasising particles

[24] This is the expected form; the actual attested form is *·relcset*, cf. section 2.2.3.

[25] Consider, for example, 3p. *-som*, *-sam* vs. *-sem*, *-sium*, *-seom* (Stifter 2006: 128).

are further discussed in section 4.9.

Code Example 4.13 – Rewriting ə in closed, unstressed syllables (only applicable with endings following stem consonants). Snippet of `10_se_orth_end_vow.rule` (section C.2.10 on page 202).

```
[ ə -> i    || palCons _ palCons ]       .o.
[ ə -> e    || palCons _ nonPalCons ]    .o.
[ ə -> {ai} || nonPalCons _ palCons ]    .o.
[ ə -> a    || nonPalCons _ nonPalCons ]
```

4.6.4 Syncope

4.6.4.1 Isolating the right vowels

Formulating rules to deal with syncope and its concomitant changes to consonant quality was the most challenging part of the lower-level rule framework. A contextual replace rule had to be created that only targets vowels in even-numbered syllables, but excludes any final syllable. After experimenting with various contextual formats, it turned out that there is actually a surprisingly intuitive way of capturing this stress-related phenomenon (cf. Code Example 4.14). The crux is that a left-most start-of-string marker (`.#.`) has to be defined for the first stressed syllable of the monolithic stem (the syllable immediately following the proclitic juncture). Allowance has to made, additionally, for words starting with either a vowel or a consonant, since a consonant (or consonant group) is optional at the start of the monolithic stem.

The rest of the rule is rather straightforward: apart from the optional initial consonant (group), if a vowel is followed by a consonant or consonant cluster, a vowel must follow as part of the next (second) syllable. This vowel is subject to syncope (deletion). The same pattern can be applied to the fourth syllable, for which—again counted from the start of the word—a triplet of vowels and consonants must have occurred. For the rare cases consisting of seven or eight syllables (counting from the stressed syllable!), five string combinations of vowels and consonants must have occurred, followed by a vowel in the sixth syllable. Note that the formulated rule does not affect vowels in final syllables by virtue of the right rule-context, which specifies that at least one consonant and a vowel must follow a syncopated vowel, i.e., *n*+1 syllable where *n* is the syllable in which a vowel is subject to syncope.[26]

[26] As mentioned in section 2.2.5, emphasising particles or *notae augentes* do not interfere with the syllable count important for syncope. Emphasising particles are initially encoded with a preceding hyphen, which never matches the right context in the syncope-rule and therefore does not lead to vowel deletion in even-numbered syllables immediately preceding an emphasising particle. In a final stage hyphens are (optionally) deleted.

Code Example 4.14 – The syncope-rule in the finite-state implementation. The operator `@->` is used to only consider the longest string of consecutive vowels. Snippet of `6_se_phon_syncope.rule` (section C.2.6 on page 200).

```
regex [
Vow+ @-> "[" ... "]" ||
.#. Cons* [ Vow+ Cons+       |
            [Vow+ Cons+]^3     |
            [Vow+ Cons+]^5 ] _ Cons+ Vow
] .o.
```

```
[ "[" Vow+ "]" -> 0 ]  .o.
```

The syncope-rule does not delete vowels immediately. One may, after all, encounter a situation where a vowel in an even-numbered syllable does not go. The rule in Code Example 4.14 first isolates the string of vowels in even syllables by enclosing them between square brackets, where ... stands for the vowel(s), e.g., `marb[ā]^M'e`, prs. ind./subj. 1pl. abs. (‘we (may) kill’) or (still rather abstract) `marb[ā^V]^D'e`, prs. ind./subj. 3pl. abs. rel. (‘who (may) kill’, ‘that (they) (may) kill’). Relative 3pl. forms are a good example of forms where syncope is optional, e.g., both *marbaite* and *marbtae*/*marbdae* are possible (cf. example (13) on page 25) and should be generated. The same goes for (not necessarily relative) 3pl. passive forms; non-syncopated (non-rel.) *do·léicetar* (as opposed to *do·léic*†*ter*) has come up during testing in the context of a case-study using the text *Táin Bó Fraích*, in Chapter 5. This form is found under the W2a lemma *do·léici* in Table 5.2 on page 111, spelled *Dolléicetar* (for the spelling *ll-* cf. section 4.8.2).

How does one get from an abstract string of the type `marb[ā^V]^D'e` (prs. ind./subj. 3pl. abs. rel.) to the surface forms *marbaite*, *marbtae* or *marbdae*? Syncope is applied by deleting the string that consists of bracketed vowels (last subrule in Code Example 4.14). In some way this rule must be circumvented to allow for alternative forms with non-syncopated vowels. For abs. 3pl. relative forms (including passives), this is implemented by encoding a surface-level ‘trigger’ vowel `^V` before the ending (cf. Code Example 4.12). Whenever this symbol follows a to-be-syncopated vowel, the rule in Code Example 4.15 kicks in. This rule optionally deletes the square right bracket of the enclosed combination of vowel and `^V`, optionally creating a subsequent string like `marb[ā^V^D'e`, which is not liable to the final rule in Code Example 4.14. After getting rid of `^V` and applying the final rule in Code Example 4.14, one is left with both `marb^D'e` and `marb[ā^D'e`. All that remains to be done is to clean up the `[`’s on the lower level, and rewrite `^D` to `t` or `d` (or both), depending on whether a vowel or consonant precedes them.[27]

[27] Ignoring, for the moment, consonant quality assimilation, (section 4.6.4.3), and vowel rewrite rules.

Code Example 4.15 – A replace rule to create alternants with the vowel marked for syncope remaining. The left bracket in the rule context is, strictly speaking, redundant. Snippet of `6_se_phon_syncope.rule` (section C.2.6 on page 200).

```
[ "]" (->) 0 || "[" Vow "^V" _ ] .o.
```

4.6.4.2 Phonotactic restrictions on syncope

In some cases syncope is never applied. Experimenting with augmented forms of weak verbs that have *ro* in stressed position pointed at many phonotactically impossible consonant clusters: *·rom*†*rba-* (*marbaid*), *·roibr*†*ss* (*brissid*), *·adr*†*lla-* (*ad·ella*). Two tentative rule contexts were formulated, either with the non-syncopated vowel before *r* or after *r*, as shown in Code Example 4.16. The rule is similar to Code Example 4.15, except that square brackets are invariably deleted, which results in only ‘non-syncopated’ forms to be eventually contained in the surface level of the FST.

Code Example 4.16 – Phonotactic restrictions on syncope: deleting square brackets to allow for vowels marked for syncope to invariably remain in surface-level forms. Snippet of `6_se_phon_syncope.rule` (section C.2.6 on page 200).

```
[ "[" -> 0 , "]" -> 0 ||
_ (Vow+ "]") r (') Cons ,
Cons r (') ("[" Vow+) _ (Vow+ "]") [l|s|"^S"]
] .o.
```

4.6.4.3 Syncope and consonant quality assimilation

As pointed out in section 2.2.3, syncopated vowels have an effect on the quality of adjoining consonants. A lost front vowel results in the new consonant cluster becoming palatal, while a lost back vowel results in a non-palatal cluster. According to McCone (1997: 6), ‘[t]his assimilation usually favours the first member of the group’, and this observation has been implemented accordingly: when consonants come to stand next to each other, the quality is transferred in a rightward, progressive fashion. For correct step-wise automatic application of syncope, it is crucial that consonant quality is encoded before the syncope process applies (which it is). Palatalisation markers (`'`) have been briefly mentioned in section 4.6.3 in relation to forms such as `marbāθ'`, where the marker is encoded as part of the ending and serves as a context for the insertion of the vowel *i* between a back vowel (representing ə) and a palatal consonant.

As said above, marking of consonant quality (or non-marking in the case of non-palatal consonants) is crucial for arriving at the right quality of consonants and consonant clusters after syncope. As Code Example 4.17 shows, the first step is to palatalise consonants when they occur before a front vowel (`frontVow` contains `e`, `ē`, `i` and `ī`) or follow `i`, `ī`. When internal vowels have been syncopated, non-palatal consonants following palatal ones are palatalised (markers added), and palatal consonants following non-palatal ones are de-palatalised (markers deleted). The right-most variable nonPalCons in the second, 'de-palatalisation' rule in Code Example 4.18 should be understood in terms of the following palatalisation marker (') since a palatal consonant is defined as a non-palatal one plus a palatalisation marker. The cascade resulting in *·aidled* (impf./past. subj. 3sg. conjunct of *ad·ella*), subject to palatalisation of the cluster *-dl-* after syncope, including subsequent vowel and consonant rewrite rules, is thus: adellāθ → ad'ellāθ → ad'llāθ → ad'l'l'āθ → ad'l'l'eθ → aid'l'l'eθ → aidlleθ → aidleth/aidled.

As already referred to in section 4.3.4, I currently operate with both abstract symbols and orthographical symbols. A better and more elegant solution to deal with a non-lenited *l* would be to unambiguously treat and encode this as the phoneme L, rather than the digraph ll, which becomes l in environments such as in the example above. Possible improvements of this nature are planned for a subsequent implementation phase.

Code Example 4.17 – Palatalisation rule based on adjoining vowel. Snippet of 4_se_phon_pal.rule (section C.2.4 on page 199).

```
[..] -> ' || nonPalCons+ _ nonPalCons* frontVow ,
[i|ī] nonPalCons+ _ nonPalCons* [ Vow | .#. | "-" ]
```

Code Example 4.18 – Post-syncope consonant quality assimilation. Snippet of 7_se_phon_cons_qual_assim.rule (section C.2.7 on page 201).

```
[ [..] -> ' || palCons nonPalCons+ _ nonPalCons* Vow ] .o.
[ ' -> 0 || nonPalCons palCons* nonPalCons _ ]
```

4.6.4.4 Analogy

A rule-based framework naturally aims at regular and predictable processes. The present work operates with Old Irish surface-level monolithic bases (rather than prehistoric roots and diachronic derivation processes, justified in section 2.4.1), in order to facilitate a trivial and predictable stem-and-ending approach. While such a synchronically oriented method takes away a great extent of the complexities and unpredictability relative to a diachronically oriented model, a certain degree of overgeneration and overgeneralisation is unavoidable. Idiosyncratic, i.e. verb-specific, processes within Old Irish, due to analogical forces, are almost impossible to capture in a framework entirely based on pre-defined stems and regular inflectional rules. Irregularly applied syncope, or the absence of syncope, is an example of such an analogical process that in the current framework cannot be satisfactorily dealt with. Analogy is often of the intra-paradigmatic type. For example, there is analogical confusion between deuterotonic and prototonic 3pl. passive forms. This can be illustrated with dependent 3pl. pass. *·epertar* (expected **·ep*†*retar*) of the strong verb *as·beir* 'says' (*ess-ber-*), influenced by deuterotonic *as·ber*†*tar* (McCone 1997: 81). Sometimes the 3sg. form influences the 3pl., e.g., prt. 3sg. pl. *·adallsat* of W1 verb *ad·ella* 'approaches, visits' for expected **·aid*†*lesat*, based on the 3sg. *·adall*. Indeed, the FST generates the expected rather than the attested form in this case.[28] Section 5.6.2 will discuss the form *celebraid* 'bids farewell' found in *Táin Bó Fraích* (Meid 1974). This form is borrowed from Latin *celebrare*, and undoubtedly for analogical reasons resistant to syncope (one would expect **ce(i)lbrid* based on strictly applied syncope rules).

4.6.5 A worked-out example: the *f*-future

Apart from the stem vowel `ā` or `ī`, a weak verb's paradigm contains three additional stems, which are formed by a suffix containing a consonant: the *f*-future, *s*-preterite and the preterite passive. The `lexc` sublexicon `weakStemFormation` in Code Example 4.11 (line numbers 120–129) has illustrated the encoding of these tense/mood stems on the lexical and surface level. This section illustrates the surface-level rewrite rules to arrive at the correct orthographical forms using the *f*-future as an example.

The future tense suffix is *if*, with syncopation of the *i* if it occurs in an even-numbered syllable. This means that this suffix often surfaces as *f*, typically with simple verbs with a one-syllable root.[29] Moreover, per the observations in section 4.6.4.3, a syncopated front vowel results in surrounding consonants becoming palatalised (if not already palatal), while a back vowel does the opposite (if not already non-palatal). However, with W1 (*ā*) verbs, one finds orthographical variants that do not show this palatalisation. In other words, with W1 verbs, at least for modelling purposes, *ā* is optionally rewritten to *i* before the future stem suffix, as illustrated in Code Example 4.19.

Code Example 4.19 – Surface-level rules for *f*-future stem formation. Snippet of `2_se_phon_non_pres_stem_form.rule` (section C.2.2 on page 198).

```
# e.g. marbā-iF -> marbi^F, marbā^F, léicī^F -> léici^F
[ ā (->) i , ī -> i || _ "^F" ] .o.
```

An example of an intermediate surface-level form, resulting from the `lexc` concatenation architecture, is fut. 3sg. abs. `marbā^Fθ'` (the same ending `θ'` as with prs. ind./subj., cf. Table 4.3). Two rules are necessary, as shown in Code Example 4.19. The fact that one set of endings (that is, the *a*-endings) both serves the present subjunctive and the future, a rule is necessary to insert a vowel after `^F` before endings starting with a consonant (e.g, `marbā^Fθ'` → `marbā^Fəθ'`).[30] This is not necessary with the subjunctive, which has no stem consonant: the

[28] Cf. the poems of Blathmac, l. 72 (Carney 1964). A similar form *Tadallsat* from *do·aidlea* (*to-ad-ell*) is found on l. 49. For textual notes relating to these forms cf. Carney (1964: 115).

[29] But not with dependent augmented forms such as (expected) **ní·reilc*___if___*ea* '(s)he will not let'.

[30] With the *s*-preterite, and with the (non-prt.) passive endings, `ə` is encoded as part of the ending set, e.g., prt. 1pl. abs. `léicī^Sə^M'i` → `léicsimmi`, prs. ind. pass. 3sg. `marbāθ'ər'` → `marbthair`. With the abs. relative 3pl. *a*-ending and pass. 1 ending the trigger `^V` is used for a mapping to `ə` after the future stem-consonant `^F`. Cf. `se.lexc` section C.1.3 on page 184 and `6_se_phon_syncope.rule` section C.2.6 on page 200.

endings follow *ā* or *ī*, which in turn demands changes to or deletion of stem vowels, rather than insertion of an ending vowel (e.g., `marbā-θ'` → *marbaith/-d*, as illustrated in Table 4.3). The alternative form with a palatalised consonant cluster due to syncope of *i*, *mairbfid*, is derived from `mar'b'i^F'əθ'`, with subsequent application of syncope (section 4.6.4). For both alternants additional intermediate orthographical rewrite rules apply, including ones that are applicable to syllables outside the stem consonant context (*m**ai**rbf-* in case of palatalised consonant cluster *-rbf-*) and to the underspecified symbol (`^F` → `f`).

In many cases, especially with prototonic forms of compounds, the syncopated vowel is not *i* (or *ā*) preceding the future-stem consonant *f*, but the root vowel, as with *ad·ella* with root *ell-*. This causes divergent inflectional patterns with independent/deuterotonic and dependent/prototonic bases; for example, one expects fut. 1pl. independent *ad·e(i)llfem* (or, perhaps, *ad·ellfam*) but dependent *·aidlibem* (or, perhaps, *·aidlebam*). The cascade of surface-level rewrite rules for *ad·eillfem* and *aidlibem* are given in Table 4.4 and Table 4.5, respectively.[31] As pointed out by Stifter (2006: 282), 'the *f* /f/ of the suffix may appear as voiced *b* (/β/) in absolute *auslaut* or intervocalically', and this has been implemented accordingly, as can be seen in step (7) in Table 4.5. Note that the preverb *ad* in the deuterotonic form is not part of the monolithic stem, and, consequently, not part of the stem-and-ending lexicon (`se.lexc`). Section 4.7.2 shows how a pretonic preverb is united with its relevant monolithic stem.

Table 4.4 – Cascade of surface-level rules and intermediate strings for deuterotonic fut. 1pl. conj. *·ellfam*/*·eillfem* (*ad·ella* 'approaches, visits').

<table>
<tr><th></th><th>ā^F</th><th>i^F</th><th>Result of rules ...</th></tr>
<tr><td>(1)</td><td>ellā^Fəμ</td><td>elli^Fəμ</td><td>cf. Code Example 4.19</td></tr>
<tr><td>(2)</td><td></td><td>el'l'i^F'əμ</td><td>add palatalisation markers, cf. Code Example 4.17</td></tr>
<tr><td>(3)</td><td>ell^Fəμ</td><td>el'l'^F'əμ</td><td>syncope, cf. Code Example 4.14</td></tr>
<tr><td>(4)</td><td></td><td>eil'l'^F'əμ</td><td>[..] -> i || backVow|e|é _ palCons</td></tr>
<tr><td>(5)</td><td>ell^Faμ</td><td>eil'l'^F'eμ</td><td>ə -> e || palCons _ nonPalCons .o.
ə -> a || nonPalCons _ nonPalCons</td></tr>
<tr><td>(6)</td><td></td><td>eill^Feμ</td><td>' -> 0</td></tr>
<tr><td>(7)</td><td>ellfam</td><td>eillfem</td><td>μ -> m .o. "^F" -> f</td></tr>
</table>

[31] The replace rules are (Appendix C Code): `2_se_phon_non_pres_stem_form.rule` (section C.2.2 on page 198), `4_se_phon_pal.rule` (section C.2.4 on page 199), `6_se_phon_syncope.rule` (section C.2.6 on page 200), `8_se_phon_stem_vow.rule` (section C.2.8 on page 201) `9_se_orth_vow_pal_cons.rule` (section C.2.9 on page 202), `10_se_orth_end_vow.rule` (section C.2.10 on page 202), `12_se_del_pal_markers.rule` (section C.2.12 on page 203) and `13_se_phon_orth_cons.rule` (section C.2.13 on page 203).

Table 4.5 – Cascade of surface-level rules and intermediate strings for prototonic fut. 1pl. conj. *·aidlebam*/*·aidlibem* (*ad·ella* 'approaches, visits').

<table>
<tr><th></th><th>ā^F</th><th>i^F</th><th>Result of rule(s) ...</th></tr>
<tr><td>(1)</td><td>adellā^Fəμ</td><td>adelli^F'əμ</td><td>cf. Code Example 4.19</td></tr>
<tr><td>(2)</td><td>ad'ellā^Fəμ</td><td>ad'el'l'i^F'əμ</td><td>add palatalisation markers, cf. Code Example 4.17</td></tr>
<tr><td>(3)</td><td>ad'l'l'ā^Fəμ</td><td>ad'l'l'i^F'əμ</td><td>syncope, cf. Code Example 4.14</td></tr>
<tr><td>(4)</td><td>ad'l'l'e^Fəμ</td><td></td><td>ā -> e || palCons _ nonPalCons</td></tr>
<tr><td>(5)</td><td>aid'l'l'e^Faμ</td><td>aid'l'l'i^F'eμ</td><td>ə -> e || palCons _ nonPalCons
.o.
ə -> a ||
nonPalCons _ nonPalCons</td></tr>
<tr><td>(6)</td><td>aidlle^Faμ</td><td>aidlli^Feμ</td><td>' -> 0</td></tr>
<tr><td>(7)</td><td>aidlebam,</td><td>aidlibem,</td><td>l -> 0 || d _ l .o.</td></tr>
<tr><td></td><td>aidlefam</td><td>aidlifem</td><td>μ -> m .o. "^F" (->) b ||
Vow _ .o. "^F" -> f</td></tr>
</table>

4.6.6 Stem entries for strong verbs

While strong verbs are outside the remit of this thesis, incorporation of this class of verbs in the current FST architecture has been anticipated. This subsection shows how stem entries are arrived at using Schumacher (2004) for reconstructed stems and Green (1995) for the Old Irish paradigms. Other important sources are Pedersen (1909–13), Thurneysen (1946), Strachan (1949), Wodtko (2007), McCone (1994), McCone (1994) and Stifter (2006). Illustrated in Code Example 4.20 is the encoding of stem entries for the strong simple verb *gaibid* 'seizes', classified as present stem type S2. The full paradigm is given in Appendix B, pages 170–171.

McCone (1997: 31) describes the S2 class as having 'palatal quality of the root-final consonant throughout, allowing for the occasional distorting effects of post-syncope progressive assimilation of quality'. As can be seen in Code Example 4.20, I do not encode the stem as *gaib*, but use the stem vowel `ī` or `ā`, which results in a surrounding palatal or non-palatal consonant (cluster) following from the progressive assimilation rules in Code Example 4.18.[32] This echoes the pre-forms for Irish given by Schumacher (2004: 318), who lists present stem *gab-i̯e/o-*, subjunctive stem **gab-ăse/o-*, future stem **géb(ā)-*, preterite stem **gab-ass-* and preterite passive stem **gab-ato-*. Prs. ind. 3sg. *gaibid*, not subject to syncope, will be produced thus: `gabīθ'` → `gabiθ'` → `gaibiθ'` → `gaibiθ` → `gaibid`.

[32] And `gabī-` → `gaib-`.

Code Example 4.20 – Monolithic stem entries for the S2 verb *gaibid*.

```
@P.W2a.ON@+gab+VROOT+S2+PRS+IND:gabī    weakPresIndEndings;
@P.W2a.ON@+gab+VROOT+S2+IMP:gabī        weakImpEndings;
+gab+VROOT+S2+IPF:gabī                  secEndings;
+gab+VROOT+S2+PRS+SUBJ:gabā             aEndings;
+gab+VROOT+S2+PAST+SUBJ:gabā            secEndings;
+gab+VROOT+S2+FUT:gébā                  aEndings;
+gab+VROOT+S2+COND:gébā                 secEndings;
+gab+VROOT+S2+PRT:gabā^S                sPretEndings;
+gab+VROOT+S2+PRT+PASS:gabā             pretPassEndings;
```

Note that *gaibid* behaves exactly like W2a in the prs. ind.,[33] imp. and ipf.; for the first two stems the accompanying weak ending sets can be used. The present `lexc` infrastructure is geared towards weak verbs, whose stems can be concatenated with a predictable set of stem consonants defined in a single continuation class `weakStemFormation` (Code Example 4.11). This means that duplicate stem entries are necessary for strong verbs. However, changing the concatenation structure is trivial, so that only one entry for each stem type would need to be keyed in; e.g., only one ‘subjunctive’ continuation class could be specified for the subjunctive stem entry `gabā`, with a subsequent continuation class `aEndings` that incorporates the continuation class `secEndings` (past subjunctive forms invariably take the secondary endings). For simple verbs like *gaibid*, and for most other strong simple verbs, five stem entries are necessary. If one includes augmented stems, or in the case of compounds, twice this amount might be needed.

The strong verb *gabaid* discussed here is probably a relatively easy example. Other strong verbs such as *beirid* and *do·beir* with root *ber* are more challenging to implement due to non-concatenative stem formation (ablaut) and fluctuation in stem-final consonant quality; cf. sg. conj. forms in the prs. ind. and prt. active paradigm in Appendix B, pages 162–163 (*beirid*) and pages 166–167 (*do·beir*). An unresolved question is whether it is in fact more economical to create full-form lexicons for these two verbs, rather than encoding complex stem formation processes, especially when parts of the paradigm additionally consist of suppletive stems, as is the case with these verbs.

4.7 Verb stem dependencies and endings

4.7.1 `se.lexc`: absolute vs. conjunct endings

The most significant rules accompanying the stem-and-ending lexicon (`se.lexc`) are those that filter out the suffixation of the wrong inflectional ending sets: the absolute and conjunct endings. Not only do flag diacritics (cf. section 4.3.5) prove convenient for handling the permitted

[33] Except for the prs. ind. 3sg. conj. zero-ending, which in my implementation has been encoded as stem minus stem vowel (`ī`).

concatenation of pretonic elements with compounds and simplexes (cf. section 4.7.2), they can simultaneously be employed to restrict the concatenation of inflectional ending sets allowed for these verb types. The stem entries for the verb *léicid* 'lets' and *do·léici* 'lets go' are contrasted in Code Example 4.21 and Code Example 4.22, respectively. The relevant filter rules, which always pertain to the upper, lexical level, are given in Code Example 4.23.

Code Example 4.21 – Monolithic stem entries for the simplex *léicid*, part of `simpleW2a.txt` (section C.4.4 on page 224).

```
+lēc+VROOT:léic            W2a;
+ro+AUG+lēc+VROOT:reiləc   W2a;
```

Code Example 4.22 – Monolithic stem entries for *do·léici*, part of `compoundW2a.txt` (section C.4.2 on page 223).

```
@R.PV.TO@+lēc+VROOT:léic                 W2a;
@R.PV.TO@+ro+AUG+lēc+VROOT:reiləc        W2a;
@D.PV@+to+PV1+lēc+VROOT:teiləc           W2a;
@D.PV@+to+PV1+ro+AUG+lēc+VROOT:tarələc   W2a;
```

Code Example 4.23 – Rules pertaining to the lexical level to filter out illegal concatenation of absolute and conjunct endings. Snippet of `1_se_filters.rule` (section C.2.1 on page 197).

```
~[ $[ ["+PV1"|"+AUG"|"+IMP"] ?* "+ABS"] ] .o.
```

```
~[ ~$["@D.PV@"] & $["+ABS"] ] .o.
```

While simple verbs may get absolute endings, compound verbs cannot. Ending sets naturally occur late in the concatenation architecture, with preceding continuation classes resulting in multiple routes or 'splits' (simplex/compound and tense/mood, and the continuation classes for the latter's ending sets). Moreover, since proclitics are in a separate lexicon, no assumption is made in the stem-and-ending lexicon as to the presence of a pretonic element with simplexes; they may take both absolute and conjunct endings. For compounds, the inflectional ending is invariably conjunct, regardless of whether the verb is independent or dependent. To avoid what would otherwise result in a proliferation of flag diacritics across the `lexc`-file, rather trivial upper-level rules based on already-existing flags (Code Example 4.23) were used to filter illegal endings out of the network.

First a rule needs to be defined that only selects those bases that do not allow absolute endings. As both prototonic bases and simplexes are accompanied by a `@D.PV@` flag, using this flag would include the simplexes, which should receive, at least optionally, all the absolute endings. The three upper-level multicharacter symbols (tags) that are uniquely found in forms that can only be accompanied by conjunct endings (that is, not absolute endings) are `+PV1`, `+AUG` and `+IMP`.[34]

[34] There is obviously no absolute ending set for imperative forms. However, while imperatives invariably carry conjunct endings, the current implementation employs a separate continuation lexicon with passive endings (`pass1Endings`), which do contain absolute endings; hence the inclusion of the imperative tag here, to restrict concatenation of absolute passive endings with imperatives.

The flag `@R.PV.TO@` equals a ‘deuterotonic’ entry, as it only allows concatenation of a preceding (matching) preverb, in this case *do* (cf. Code Example 4.5). As both simplexes and prototonic bases are accompanied by `@D.PV@`, the complement set of forms with the latter flag solely consists of deuterotonic monolithic stems (with the `R` flag), which cannot take absolute endings.[35]

4.7.2 Flag diacritics, lexical verb type and dependency

Section 4.7.1 has already illustrated how preceding flags are used to distinguish between simplexes and prototonic bases on the one hand, and deuterotonic ‘stems’ on the other. Although this binary division does not correspond to either a lexical verb type (simple/compound) or dependency (`@D.PV@` has no effect on the endings suffixed to simple verbs), it was found to be useful in relation to two major restrictions:

1. Both simple verbs and prototonic bases cannot be preceded by a pretonic lexical preverb (`@D.PV@`, i.e., disallow a preverb); and conversely,
2. the monolithic stem of deuterotonic compounds *must* be preceded by a pretonic lexical preverb (`@R.PV.X@`, require a preverb, with `X` = specific preverb).

The `@D.PV@` flag only blocks a pretonic lexical preverb. This entails a further three restrictions, with the final two being catered for by upper-level tag filters:

- The meaningless particle *no* and compound verbs cannot co-occur (`@D.PART.NO@` accompanies compound stems, as shown in Code Example 4.11).
- No imperative particles with non-imperative forms, and no imperative forms with proclitics other than *no* (with simple verbs) and the imperative particle.
- No pretonic *ro* with compound verbs.

Code Example 4.24 illustrates how flags are encoded in the network with a random selection of verbs.

The upper-level tag `DEUT` is added in the final stage to aid subsequent morphosyntactic disambiguation (not part of this thesis) should the monolithic stem be separated by space from the pretonic preverb. The parses/analyses in Code Example 4.24, resulting from `apply up` (or, simply, `up`), constitute a selection of possibilities: relative readings and possible mutations not shown in the spelling (cf. section 4.8.2) are left out for purposes of clarity. The last two forms, **do(·)mairbfea* and **ad(·)brisiu*, are produced by telling the compiler to ignore the flags.[36]

[35] Formulating the complement language here is much more convenient that the alternative, which is to specify the restriction in terms of individual pretonic preverbs, each with a different `R` flag.

[36] In the current implementation, the flag diacritic is eliminated which cancels the require restriction and *does* result in these separated deuterotonic stems (with the added tag `+DEUT`) to be part of the network. When a text is not likely to have, say, *do* and *léici* separated out, the flag can be left in, the accompanying restrictions causing a form such as *léici* (from *do·léici*) not to come up with the ‘deuterotonic’ interpretation (reducing unnecessary ambiguity).

Code Example 4.24 – Illustrating flag diacritics with a random selection of verb forms.

```
foma[1]: up léicit
@D.PV@ l ē c +VROOT +W2a +PRS +IND +ABS +3P +PL
@D.PV@ l ē c +VROOT +W2a +PRS +SUBJ +ABS +3P +PL

foma[1]: up léicea
@D.PV@ l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +SG
@D.PV@ l ē c +VROOT +W2a +PRS +SUBJ +ABS +1P +SG
DEUT + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +SG

foma[1]: up doléici
@P.PV.TO@ t o +PV1 +PROCL_JUNCT @R.PV.TO@ + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +SG
@P.PV.TO@ t o +PV1 +PROCL_JUNCT @R.PV.TO@ + l ē c +VROOT +W2a +PRS +IND +CONJ +2P +SG

foma[1]: up nommarbat
@P.PART.NO@ n o +CONJ_PART +PRON +A +1P +SG +PROCL_JUNCT @D.PV@ + m a r b +VROOT +W1 +PRS +IND +CONJ +3P +PL
@P.PART.NO@ n o +CONJ_PART +PRON +A +1P +SG +PROCL_JUNCT @D.PV@ + m a r b +VROOT +W1 +PRS +SUBJ +CONJ +3P +PL
@P.PART.NO@ n o +CONJ_PART +PRON +A +1P +SG +PROCL_JUNCT @D.PV@ + m a r b +VROOT +W1 +IMP +CONJ +3P +PL

foma[1]: set obey-flags OFF

foma[1]: up domairbfea
@P.PV.TO@ t o +PV1 +PROCL_JUNCT @D.PV@ + m a r b +VROOT +W1 +FUT +CONJ +3P +SG

foma[1]: up adbrisiu
@P.PV.AD@ a d +PV1 +PROCL_JUNCT @D.PV@ + b r i s +VROOT +W2a +PRS +IND +CONJ +1P +SG
```

4.7.3 Rationale for stem entries on a per-verb/lemma basis

The employment of flag diacritics and the approach whereby the stem-entry procedure is on a per-verb basis is an important choice in the present work, but also an arbitrary one. There are certainly other ways to organise the stem entries and encode restrictions and dependencies. Flag diacritics could be done away with and every (pretonic) preverb could be combined with every root if upper-level filters were defined specifying the various dependencies regarding those preverb-and-stem combinations. And one could perhaps implement stems classified according to root, as illustrated in Code Example 4.25, as such catering for more than one strong verb with root *ber*.

Code Example 4.25 – Part of a hypothetical stem-entry list for *ber* if classifying stem entries by root.

```
+ber+VROOT+S1+PRS+IND:ber
+ber+VROOT+S1+PRT:bert
+ro+AUG+ber+VROOT+S1+PRT:rubart
```

However, managing a list of all the potential combinations and dependencies in a separate list would be quite cumbersome when pretonic prefixes or (potentially stressed) infixes such as *ro* are present, especially with strong verbs. For example, the simplex *beirid* ‘carries’ does not allow an augment in combination with its root as it employs the suppletive stem *ucc*. The verb *as·beir* ‘says’, with the same root *ber*, does allow augmentation (with *ro*, e.g., aug. prt. 3sg. *as·rubart* ‘has said’), whereas *do·beir* (also with root *ber*) shows—like the simplex *beirid*—suppletion in this case, but with two suppletive stems: *do·rat* ‘has given’ and *do·uic* ‘has brought’ (cf. Appendix B, pages 168–169).[37] Flags in this case could offer a solution, as shown in Code Example 4.26.

Let us focus again on the weak verbs which are at the core of my implementation. In Code Example 4.21 and Code Example 4.22 I contrasted the stem entries necessary for *léicid* and *do·léici*, respectively. Analogous to the ‘experiment’ with verb forms with root *ber*, the stem-entry lists of the simplex and compound entries could be conflated into the list in Code Example 4.27. I added to this list two non-augmented monolithic stems for *as·oilgi* ‘opens’, constituting another compound with verb root *lēc* (the historical derivation *uss-od-lēc-* is taken from Stifter (2006: 364)).

[37] For the reconstruction of the composite elements in *do·ratai* cf. Schumacher (2004: 266). For encoding complications with non-palatal final stem consonants before `ī` with W2 verbs, cf. section 5.6.2.

Code Example 4.26 – Part of a hypothetical stem-entry list for *ber* if classifying stem entries by root, adding flag diacritics.

```
! simplex
+ber+VROOT+S1+PRS+IND:ber
@D.PV@+ucc+VROOT:ucc
@D.PV@+ber+VROOT+S1+PRT:bert

! compounds (do·beir, as·beir, etc.)
@R.PV.ESS@+ber+VROOT+S1+PRT:bert
@R.PV.TO@+ber+VROOT+S1+PRT:bert

! as·beir
@R.PV.ESS@+ro+AUG+ber+VROOT+S1+PRT:rubart

! do·beir
@R.PV.TO@+ro+AUG+ad+PV2+dā+VROOT:rat
@R.PV.TO@+ucc+VROOT:ucc
```

Code Example 4.27 – Monolithic stem entries for a hypothetical stem-entry list for verbs with root *lēc*.

```
! simplex
@D.PV@+lēc+VROOT:léic
@D.PV@+ro+AUG+lēc+VROOT:reiləc

! compounds
! do·léici
@R.PV.TO@+lēc+VROOT:léic
@R.PV.TO@+ro+AUG+lēc+VROOT:reiləc
@D.PV@+to+PV1+lēc+VROOT:teiləc
@D.PV@+to+PV1+ro+AUG+lēc+VROOT:tarələc

! as·oilgi (non-augmented)
@R.PV.USS@+od+PV2+lēc+VROOT:oiləg
@D.PV@+uss+PV1+od+PV2+lēc+VROOT:osələc
```

If operating with flag diacritics, one cannot avoid—even in this 'conflated stem-entry' approach—having to list doublets such as `reiləcī`, which is 'shared' by both *léicid* and *do·léici* (e.g, dependent simplex aug. prt. 3sg. *ní·**reilic*** 'has not let' and independent aug. compound prt. 3sg. *do·**reilic*** 'has let go'). Admittedly, a monolithic stem of the type `reiləcī` would only have to be listed once if an upper-level tag restriction of the type '`ro+AUG` may invariably be used with `lēc+VROOT`' were added instead of employing flag diacritics. Incidentally, not employing flag diacritics and maintaining a separate list of dependencies is somewhat more straightforward in the case of simple verbs, a large amount of which use *ro* for augmentation purposes (Stifter 2006: 252).

In other words, organising stem entries by verb root rather than lemma is theoretically possible. However, this is not the *modus operandi* entertained in this work. An approach on the basis of lemmas instead was motivated by three insights:

1. A classification on a lemma basis fits the approach of using monolithic stems better: invasive changes to roots combined with preverbs (due to the stress system, cf. section 2.3) lead to divergent bases which are hard to relate back to etymologically related compound verbs which share the same verb root.[38]

2. A root-based approach fails to make a distinction between simple and compound, each of which are subject to different morphotactic restrictions/dependencies. Simplexes, for example, productively and predictably take *ro* as the augment and obligatorily take *no* to create a binary verbal complex.[39]

3. Categorising monolithic stems as in Code Example 4.21 and Code Example 4.22 allows us to arrive at a minimum or average amount of stems across verb lemmas that is necessary to capture the Old Irish verbal system with straightforward morphological rules for stems and endings (cf. section 2.4.1), which provides a diagnostic for morphological complexity. This approach will be particularly insightful and perhaps even necessary when dealing with strong verbs, and those with suppletive stems, whose stem and ending formation results in additional layers of inflectional complexity, especially in terms of choosing stem entries.[40]

4.8 Combining proclitic and stem-and-ending lexicons: upper-level tag filters

4.8.1 Concatenating the lexicons

Section 4.7.2 already discussed contiguous binary forms of the verbal complex and the role of flag diacritics to match the correct pretonic and tonic elements. The code to arrive at binary forms is shown in Code Example 4.28. Verbs in binary forms invariably carry conjunct endings. Each verb form as part of `se.lexc` that contains conjunct inflection is extracted and concatenated with `proclitic.lexc` (slightly altered after the application of some replace rules). Each proclitic element followed by the part with the monolithic stem and ending is now separated by `+PROCL_JUNCT` on the lexical level and '·' on the surface level. The variable

[38] Although not provable, it is also unlikely, due to the sheer amount of inflectional forms associated with combinations of preverbs and lexical roots, that Old Irish speakers invariably generated 'on the fly' inflected forms based on underlying roots.

[39] Note that in the current implementation, a single flag `@D.PV@` is sufficient in the first lexicon of `se.lexc` (Code Example 4.11) to cover all simple verbs; this flag therefore does not have to accompany any subsequent simple verb stem entry.

[40] Consider *beirid* and compounds with verb root *ber* in Appendix B, which show complex stem and ending formation (ablaut and fluctuation in stem-final consonant quality).

LEXseConjWithMut points to a preceding step, taking care of consonant mutations occurring across the proclitic boundary. The addition of initial consonant mutations is discussed in section 4.8.2.

Code Example 4.28 – Concatenating (derivatives of) proclitic.lexc and se.lexc, defined as variables. Snippet of procl_se.script (section C.3.10 on page 215).

```
define LEXunfiltered [LEXprocl "+PROCL_JUNCT":· LEXseConjWithMut
    ] ;
```

4.8.2 Matching mutation tags and underspecified orthography

A benefit of operating with two separate lexicons for proclitics and stem-and-endings is the relatively easy coverage of possible mutations caused by (mostly) proclitic elements. After (grammatically correct) forms with absolute and conjunct endings have been extracted from the se.lexc file (cf. section 4.7.1), all possible initial consonant mutations are added relevant to absolute and conjunct verb forms. Apart from encoding consonant mutations on the surface level, it was found that a lexical-level tag was linguistically motivated (mutations have grammatical significance). It also serves an important purpose when combining the proclitic and stem-and-ending lexicon.

Each mutation is defined as a twofold operation with a prefixed two-level relation consisting of a mutation tag on the upper level followed by a replace rule targeting the lower level of this prefixed transducer (Code Example 4.29). Initial consonant mutations (mostly) apply to deuterotonic compounds and dependent formations, which constitute the greater part of the subset of verb forms that carry conjunct endings. It is with these formations that one frequently witnesses an initial consonant mutation on the following stressed syllable. The prefixed mutation transducer results in a forms such as in (22); its composition with $\frac{\texttt{\textasciicircum LEN t}}{\texttt{t h}}$ results in the mapping in (23). Note that the application of initial consonant mutation is independent of whether a contiguous proclitic occurs or not; as parsing is word-based, any ‘mutated’ monolithic stem needs to be generated, regardless of an immediately preceding proclitic; this prefix might not be consecutive to the monolithic stem (which, at this stage, is unknown). In other words, all three mutations have to be applied to most forms carrying conjunct endings in the inflectional paradigm.

I should also add that ‘mutation tags’ are prefixed independently of whether the subsequent replace rule actually rewrites any lower symbol. This means that those verb forms whose word-initial consonants are either not liable to a consonant mutation, or do not show this in the spelling, also get assigned either +LEN, +NAS or +H to the lexical/upper level. This is both linguistically justified and serves to disambiguate between mutations caused by proclitics, e.g., a leniting or nasalising relative, or a mutation caused by an infixed pronoun (cf. below). Some consonant mutations are only optionally marked in the spelling; for example, as can be seen in Code Example 4.29, lenited *f* might either remain *f* (the rule should therefore only optionally apply), contain a *punctum delens* (*ḟ*) or disappear altogether (0).

Code Example 4.29 – A twofold operation to mark lenition on initial consonants using two variables (transducers). Snippet of `mutation.script` (section C.3.8 on page 213).

```
define lenTwoLevel ["+LEN":"^LEN" ] ;
```

```
define lenLower [
[ [..] -> h || "^LEN" [c|p|t] _ ] .o.
# No lenition with sc, sp, st, sm
[ s (->) ṡ || "^LEN" _ \[c|p|t|m] ] .o.
[ f (->) ḟ|0 || "^LEN" _ ] .o.
"^LEN" -> 0
] ;
```

(22)
```
+LEN + t o +PV1 + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +SG
                  ^LEN t e i l c i
```

(23)
```
+LEN + t o +PV1 + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +SG
                  t h e i l c i
```

A mutation transducer for nasalisation can be used for optional nasalisation with relative forms of simple verbs carrying an absolute ending (e.g., prs. ind. 3sg. *mbrises* or *ṁbrises* ‘that ... breaks’ alongside *brises*). This entails extracting a subset of the lexicon with absolute endings and prefixing a nasalisation transducer, and performing a union operation (`|`) with the original ‘absolute’ transducer.

Another phenomenon is captured in `mut.script` that is related to consonant mutations, namely, the double spelling of an initial consonant*l*, *n* and *r*, to mark that these consonants are not lenited (reflecting [L], [N], [R], respectively). This `nonLenAnlaut` transducer is shown in Code Example 4.30. This transducer is only prefixed to forms with the conjunct inflectional ending set, as it is after a proclitic that the doubling of initial consonants may occur, to unambiguously mark that these consonants are not lenited. No accompanying upper-level are added in this case as these digraphs reflect spelling variation; the orthographical convention is to use single *l/n/r* for these sounds (when not lenited). Code Example 4.31 shows how the final conjunct lexicon with mutations (defined as the variable `LEXseConjWithMut`) is constructed.

Code Example 4.30 – A prefixed transducer to deal with digraphs for non-lenited anlaut consonants. Snippet of `mutation.script` (section C.3.8 on page 213).

```
define nonLenAnlaut [
[ m -> {mm} , n -> {nn} , r -> {rr} , l -> {ll} || .#. _ ] ] ;
```

Code Example 4.31 – Unioning the lexicon with conjunct endings (`LEXseConj`) with one subjected to mutations and one with digraphs for non-lenited *anlaut* consonants. Snippet of `se_3_mut.script` (section C.3.13 on page 219).

```
define LEXseConjWithMut  [
[ LEXseConj .o. nonLenAnlaut ] |
[ [ mutTwoLevel LEXseConj ] .o. mutLower ] |
LEXseConj
] ;
```

All verb forms in `LEXseConjWithMut` now optionally have all the three mutations applied (on both the lexical and surface level). Combining `proclitic.lexc` and the derivative of `se.lexc` (`LEXseConjWithMut`) leads to incompatible mutation tags in, for example, aug. prt. ind. 3sg. *ra·bris* 'has broken him', where one would expect nasalisation of initial *b-* after the 3sg. masc. infixed pron. (class A) *-a*. Ignoring the incompatible combinations of upper-level mutation tags would result in wrong parses (analyses), exemplified by random instances in Code Example 4.32. Note that 'incompatible' includes binary forms whose specific 'mutation tag' is not mirrored on either side of the proclitic juncture.

Code Example 4.32 – Examples of incompatible 'mutation tags' after concatenating (a derivative of) `proclitic.lexc` (section C.1.2 on page 179) and the 'mutated', conjunct forms of (a derivative of) `se.lexc` (section C.1.3 on page 184).

```
foma[1]: up rabris
r o +AUG +PRON +A +3P +SG +NEUT +LEN +PROCL_JUNCT + b r i s
    +VROOT +W2a +PRT +CONJ +3P +SG
r o +AUG +PRON +A +3P +SG +NEUT +LEN +PROCL_JUNCT +H + b r i s
    +VROOT +W2a +PRT +CONJ +3P +SG
r o +AUG +PRON +A +3P +SG +MASC +NAS +PROCL_JUNCT +LEN + b r i s
    +VROOT +W2a +PRT +CONJ +3P +SG
```

Code Example 4.33 gives the filter rule for nasalisation, counteracting this lexical-level tag incompatibility. This rule results, for example, in a form *ra·mbris* now correctly receiving a lexical-level analysis with an infixed pronoun masc. rather than a neut. tag. In the current rule framework, the 'mutation tag' originating from entries in `proclitic.lexc` (that is, the final tag of the proclitic string) is removed in favour of the one immediately following `+PROCL_JUNCT`, reflecting the position in the verbal complex where the consonant mutation is realised. However, changing the deletion rule is trivial should the need arise to keep the tag immediately to the left of the proclitic juncture in.

Code Example 4.33 – Filtering out incompatible lexical-level nasalisation tags. Snippet of `procl_se_filter_6_mut.rule` (section C.2.19 on page 207).

```
~[ $["+NAS" "+PROCL_JUNCT" \["+NAS"] ] ] .o.
```

```
~[ $[\["+NAS"] "+PROCL_JUNCT" "+NAS"] ] .o.
```

4.8.3 The conjunct particle *no*, passives, and infixed pronouns

The dependencies in relation to the meaningless verbal particle *no*, only allowed with simple verbs, have already been addressed in section 4.4.2. With non-relative forms, the ipf., past subj. and conditional (all with secondary endings), the conjunct particle *no* is obligatory. However, outside the ipf., past subj. and the conditional, in non-relative contexts, this pretonic particle occurs only to form a slot for infixed pronouns; a form such as ***no·léicet*, with *-et* pointing to a prs. ind., prs. subj. or imp. 3pl. (conjunct) ending, is therefore illegal, as no infixed pronoun is present after *no* (and there is a special absolute relative form for 3pl.).

With relative forms of simple verbs, *no* may appear without an infixed pronoun. However, the 3sg., 3pl. and 1pl. have special absolute relative endings, in which case *no* is not employed, unless a pronoun appears, which is always infixed. In other words, *no* cannot appear with 3rd person and 1pl. relative forms, unless an infixed pronoun is present. The restrictions mentioned above translate into the upper/lexical-level tag filter rules as shown in Code Example 4.34.

Code Example 4.34 – The rule `procl_se_filter_2_no.rule` filters out incompatible lexical-level combinations with *no* (section C.2.15 on page 204).

```
regex [
# e.g. **no·reilic
~[ $[ {no} "+CONJ_PART" ?* "+AUG" ] ] .o.

# "No" (non-rel.) without inf. pronoun only with secondary
    endings (i.e. not with tenses/moods below)
~[ $[{no} "+CONJ_PART"] & ~$["+REL"] & ~$["+PRON"] &
    $["+PRS"|"+IMP"|"+FUT"|"+PRT"] ] .o.

# Relative "no" forms WITHOUT inf. pron. are restricted to those
    person/number forms that do not have a special absolute rel.,
    e.g. prs. ind. 1pl. rel. **no·léicem (> léicme), but 1pl.
    secondary end. (invariably conj.) no·léic(fi)mis (both main
    and relative)
~[ $[{no} "+CONJ_PART" ?* "+REL"] & ~$["+PRON"] & $["+CONJ"
    ["+3P" | "+1P" "+PL"] ] & $["+PRS"|"+IMP"|"+FUT"|"+PRT"] ]
] ;
```

Only 3sg. and 3pl. passive endings exist. Passive forms for other person/number combinations are realised by means of the binary verbal complex, with invariably a 3sg. pass. (conjunct) ending and an infixed pronoun of the first or second person, e.g., *dom·berar* ‘I am being given / one gives me’. The empty particle *no* is needed for simple verbs (equally only with the presence of an infixed pronoun of the first or second person), e.g., *nob·léicther* ‘ye are being let, one lets ye’. In other words, one cannot get:

1. A 3sg. pass. conjunct ending with an infixed pronoun 3sg. (the 3sg. pass. ending already denotes the subject).
2. An infixed pronoun with the pass. 3pl. ending.
3. Following from 2.: *no* with a pass. plural.

This translates into the restrictions in Code Example 4.35.

Code Example 4.35 – The rule `procl_se_filter_4_pass.rule` filters out incompatible lexical-level combinations with passive conjunct endings (section C.2.17 on page 205).

```
regex [
# no 3pl. pass. with inf. pron., e.g. **nob·marbtar,
    **don·léicfiter
~[ $[ [{no} "+CONJ_PART" | "+PRON"] ?* "+PASS" "+CONJ" "+3P"
    "+PL"] ] .o.

# no 3sg./pl. inf. pron. with pass. (sg.), e.g.
    **na/ra/da·léicther
~[ $[ ["+CONJ_PART"|"+PV1"|"+AUG"] ?* "+PRON" (?) "+3P" ?*
    "+PASS" "+CONJ" "+3P" "+SG"] ]
] ;
```

4.9 Emphasising particles and agreement

Code Example 4.36 illustrates how allowance can be made for different ‘suffixation routes’ for absolute and conjunct forms (for insertion of stem entries cf. section 4.6.2 and section 4.6.3). As can be seen in the code excerpt, I included the emphasising particles[41] (or *notae augentes*) and suffixed pronouns in what is otherwise a stem-and-ending lexicon file. Suffixed pronouns can only appear with simple verbs and absolute endings, and occur almost exclusively with 3sg. endings, which is reflected in the code: the continuation class `suffAbs3sg` represents a sub-lexicon specifically for absolute 3sg. inflection, which, apart from pronoun suffixation, also eventually leads to the 3sg. emphasising particle (absolute endings can only be followed by an agreeing emphasising particle). The lexicon file `se.lexc`, section C.1.3 on page 184, contains a full-form lexicon for the substantive verb, including present and preterite formations

[41] Since `ə` is already used for the surface forms of unstressed vowels in endings following stem consonants, another underspecified symbol is used for the vowel in the 3sg./3pl. particles; `^Vemph3P` is rewritten to either `o` or `a`, or to `e`, `iu` or `eo`, depending on the quality of the consonant or the vowel preceding.

with suffixed pronouns (towards the end, line numbers 384–403). Intrusive *-th-* in the preterite originates in the present *tá-*, but spreads out from there (Thurneysen 1946: 271); for example, 3sg. fem. *táthus* (‘she has’) → *boíthus* (‘she had’). Note that `LEXICON suffAbs3sg` in Code Example 4.36 results in the suffixation of either a pronoun or a particle; both is not possible.

Code Example 4.36 – A snippet of `se.lexc`, focusing in on the pronominal suffixes and emphasising particles (section C.1.3 on page 184).

```
100 LEXICON simpleW1

    +marb+VROOT:marb        W1;

114 LEXICON W1
115 +W1:ā   weakStemFormation;

120 LEXICON weakStemFormation
121 +PRS+IND:0      weakPresIndEndings;

131 !\\\\\ ENDINGS /////

133 LEXICON weakPresIndEndings

137 +ABS+3P+SG:θ'               suffAbs3sg;

147 +CONJ+3P+SG:0               Emph;

405 !!\\\\\ SUFFIXES /////
406
407 LEXICON Emph
408         #;
409 0:-     Emph2;
410
411 LEXICON Emph2
412 +EMPH+1P+SG:s^Vemph1SG          #;
413 +EMPH+2P+SG:s^Vemph2SG          #;
414 +EMPH+3P+SG+MASC:s^Vemph3Pm     #;
415 +EMPH+3P+SG+NEUT:s^Vemph3Pm     #;
416 +EMPH+3P+SG+FEM:si              #;
417 +EMPH+1P+PL:ni                  #;
418 +EMPH+2P+PL:si                  #;
419 +EMPH+3P+PL:s^Vemph3Pm          #;
420
421 LEXICON suffAbs3sg
422                         Emph3sg;
423 +PRON+1P+SG:um          #;
424 +PRON+2P+SG:ut          #;
425 +PRON+3P+SG+MASC:i      #;
426 +PRON+3P+SG+NEUT:i      #;
```

```
+PRON+3P+SG+FEM:us      #;
+PRON+1P+PL:unn         #;
+PRON+2P+PL:uib         #;
+PRON+3P+PL:us          #;
```

```
LEXICON Emph3sg
                                #;
+EMPH+3P+SG+MASC:-s^Vemph3Pm    #;
+EMPH+3P+SG+NEUT:-s^Vemph3Pm    #;
+EMPH+3P+SG+FEM:-si             #;
```

The unrestricted nature of emphasising particle suffixation with conjunct endings (reflected in the 'generic' `Emph` continuation class under `LEXICON weakPresIndEndings`) is intimately connected with the implementation strategy of this project, which is word-based parsing (a 'word' denoting a string separated by space), rather than POS tagging. The implementation strategy is to optionally deal with proclitics and the string immediately to the right of the proclitic juncture (incorporating the monolithic stem) as separate entities, separated by space. As I have pointed out in section 4.4.2, this might reflect editorial policy; conjunct particles, as well as *ro* and *no*, might be separated from the verb.

However, even without knowledge about a potentially preceding proclitic, some restrictions relative to emphasing particles are known. An absolute ending can only be followed by an agreeing emphasising particle (the `LEXICON Emph3sg` is shown in the example). This is in contrast to conjunct endings, which signify a dependent or compound form,[42] and which, by virtue of the presence of a pretonic preverb or particle, allow an infixed (rather than suffixed) pronoun, which can be any person/number combination (one has to anticipate the scenario in which a proclitic and infix are separated from the remaining part of the verbal complex). This translates into the possibility of any emphasising particle following conjunct endings, which explains why the 'general' `LEXICON Emph2` follows a conjunct ending.

4.10 Final replacements and word-initial capitals

A final script applies the union operation to the different lexicons defined so far. As noted in section 4.7.2, the lexical-level tag `DEUT` is added to the corresponding monolithic bases (e.g., `ella` of *ad·ella*), and these forms can be blocked by leaving the flag in (of the format `R.PV.X`), if so desired. A final rule takes care of the encoding of the proclitic boundary marker '·' with binary forms resulting from the concatenation of the derivatives of the proclitic and stem-and-ending lexicon (cf. section 4.8.1); this boundary marker is optionally rewritten to a hyphen, which in turn is optionally rewritten to zero. The final transducer with the combined lexicons now contains binary forms which have either the mid-high dot, a hyphen or no space for the proclitic juncture, for maximum coverage. Note that by optionally rewriting the hyphen to zero

[42] One exception is the imperative, which invariably carries a conjunct ending—even in the case of a simplex—but might be independent.

as a last step, optional forms are produced in which the inflectional ending and the emphasising particle are contiguous (that is, without a hyphen).

One addition to the transducer was made that borders on syntactic disambiguation: the upper-casing of word-initial letters in those forms that one expects can occur in clause/sentence-initial position—rather than capitalisation of word-initial lower-case letters in every single form. There are a few situations in which upper-case forms are not possible (or, at least, not expected). One such subset consists of relative forms. Another one comprises conjunct forms (separated from their proclitic element), except imperatives, which always have conjunct endings and may occur in sentence-initial position. However, remember that all possible mutations were added to (separated) forms with conjunct endings. These forms assume that a proclitic precedes. If an imperative form occurs in clause-initial position (i.e., is independent), obviously no proclitic precedes and the initial consonant mutation cannot come from anywhere (i.e., upper-casing of an initial consonant subject to a mutation is not valid in this case).

Furthermore, an independent imperative cannot be accompanied by an emphasising particle that does not agree with the verb ending; hence upper-casing these forms is incorrect. Code Example 4.37 illustrates how these restrictions work out by analysing (`apply up`) verb forms with initial consonants in both upper-case and lower-case. The output `???` means that a string is not recognised, that is, not part of the language of the Finite-State Transducer).

Code Example 4.37 – Some inflected forms illustrating the restrictions encoded in relation to capitalisation of word-initial letters.

```
foma[1]: up léicit
l ē c +VROOT +W2a +PRS +IND +ABS +3P +PL
l ē c +VROOT +W2a +PRS +SUBJ +ABS +3P +PL

foma[1]: up Léicit
l ē c +VROOT +W2a +PRS +IND +ABS +3P +PL
l ē c +VROOT +W2a +PRS +SUBJ +ABS +3P +PL

foma[1]: up léicet
l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
l ē c +VROOT +W2a +IMP +CONJ +3P +PL
H + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
H + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
H + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
LEN + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
LEN + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
LEN + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
NAS + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
NAS + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
NAS + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
```

```
DEUT + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
DEUT + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
DEUT + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
DEUT +H + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
DEUT +H + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
DEUT +H + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
DEUT +LEN + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
DEUT +LEN + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
DEUT +LEN + l ē c +VROOT +W2a +IMP +CONJ +3P +PL
DEUT +NAS + l ē c +VROOT +W2a +PRS +IND +CONJ +3P +PL
DEUT +NAS + l ē c +VROOT +W2a +PRS +SUBJ +CONJ +3P +PL
DEUT +NAS + l ē c +VROOT +W2a +IMP +CONJ +3P +PL

foma[1]: up Léicet
l ē c +VROOT +W2a +IMP +CONJ +3P +PL

foma[1]: up Léicet-sam
l ē c +VROOT +W2a +IMP +CONJ +3P +PL +EMPH +3P +PL

foma[1]: up Léicet-si
???
```

4.11 Synthesis

This chapter has provided the reader a guide into the programmatic aspects involved in modelling the Old Irish verbal complex. The computational paradigm and test set of verbs have been discussed in section 4.2 and section 4.3, respectively. The often non-trivial relation between underlying and surface forms of Old Irish verbs has been successfully tackled in the FST concatenation and rule framework.

The most significant aspect of implementation is the encoding of the verb stem using a pre-syncopated, ‘monolithic’ base (section 4.6.2) to economically and insightfully create rules to cater for straightforward ending sets. Other important aspects of the implementation include:

- Assigning root forms to the upper/lexical level and semi-surface forms to the lower level (cf. section 4.4).

- Two stand-alone lexicons for proclitics and stems-and-endings (incorporating the monolithic stem), which are combinable (section 4.4.3, section 4.8).

- Replace rules for the application of (non-)syncope (section 4.6.4).

- Encoding separated dependencies with different verb and stem types using flag diacritics (section 4.7).

Chapter 5

Case study: *Táin Bó Fraích*

5.1 Introduction

This chapter deals with preliminary testing results after applying the Finite-State (or lexical) Transducer (FST) for Old Irish verbs to the Early Irish text *Táin Bó Fraích* (TBF), edited by Meid (1974). The purpose of the case study, its background and some important statistics are discussed in section 5.2. A few minor issues with the FST were found on the basis of testing on this text, which were rectified (section 5.3). Section 5.5 provides the results, obtained by using the tool `flookup`, focusing on 27 genuinely Old Irish W1 and W2a verb lemmas and their inflected forms found in the text (cf. Table 5.2 on page 111). The recognition score is measured against four texts contained in *Fingal Rónáin and other stories* (Greene 1955) to test whether the results for *Táin Bó Fraích* are generalisable across Early Irish texts. A discussion of the results for *Táin Bó Fraích* follows in section 5.6. Some issues that proved to be not solvable in the context of this thesis are discussed in section 5.6.2. Lemmatisation of the text using an lemmatiser based on eDIL (Dereza 2016) is the subject of section 5.7. A synthesis of this chapter is provided in section 5.8.

5.2 *Táin Bó Fraích*

5.2.1 Purpose of the case study

The main goal of this case study is to apply the Old Irish Finite-State Transducer (FST) to *Táin Bó Fraích* (Meid 1974), in order to gauge the balance between weak verbs and strong verbs in an Early Irish text, and the effort involved and recognition score pertaining to successful analysis of weak verb forms. A narrative text was chosen since it provides a context to individual words. Moreover, Early Irish narrative texts provide, apart from canonical Old Irish forms, non-canonical Old Irish spellings as well as Middle Irish forms, which in our case are comprehensively documented in the notes and vocabulary sections as part of the edition by Meid (1974) and, more recently, Meid et al. (2015). The text therefore provides a good testing ground for an FST which adheres, largely, to normalised Old Irish: how does the FST—based on a few test verbs—deal with grammatical forms and spellings in 'real' texts, which hardly ever solely contain 'genuine' Old Irish forms? In sections 6.4.4 and 6.4.5 I will return to the matter with some preliminary ideas on dealing with linguistic variation.

A lemmatisation tool for Early Irish (Dereza 2016)—separate from the FST—is employed to augment the FST analysis and to gauge how much one can gain in terms of word recognition (cf. section 5.6.4). Dealing with non-standard inflections and spelling variation is only very preliminarily explored in this thesis. However, it is expected that the Early Irish Lemmatiser (and, ideally, a standardiser) is necessary even in the hypothetical case of the FST incorporating all Old Irish (verb) forms and a large amount of Middle Irish forms, due to the fact that each text shows its own idiosyncrasies. Moreover, grammatical and orthographical variation becomes ever more pronounced in Middle Irish.

5.2.2 Background to the text

The edition used is Meid (1974), the narrative text of which is on CELT.[1] I am thankful to Dr Teresa Lynn for providing me with a machine-readable version of the vocabulary in Meid (1974), which constituted the basis for a rudimentary Part-Of-Speech-tagged version of the text (Lynn 2012). As my objective is automatic morphological analysis and lemmatisation, the POS-tagged text was of no immediate relevance. Moreover, a substantial amount of words in the POS-tagged text are not accompanied by a tag due to incorrect processing of hyphenated dependent verb forms in the vocabulary. Admittedly, the goal of Lynn (2012) was not an in-depth study of TBF, but an exploration of the possibilities and benefits of applying computational methods to medieval Irish texts, and one of the first of its kind.

I will give a brief synopsis of the introductory background to TBF as contained in Meid (1974). However, Meid based himself mainly on the literary discussion of the text provided in James Carney's 1955 *Studies in Irish literature and history*. The provenance of TBF is a primitive saga which at some stage during the medieval period split into two traditions, one surviving as an oral tradition, and one that eventually developed into the written story as it has come down to us in four manuscripts (ranging from the second half of the 12th century to the 16th century).[2] The theme from which the two independent traditions arose might have been an *aided* 'death story', a love story involving a man called Fróech who was killed by a water-monster, potentially after a local Connacht tradition connected to the place-name *Dublinn Froích* 'heathery pool', and/or modern *Carn Fraoich* 'heathery mound' (FitzPatrick 2004: 60).[3]

The TBF story contains two parts, which clearly have a different provenance: part one deals with the wooing of Findabair, royal daughter of Ailill and Medb of Connacht, part two with the recapturing of Fróech's cattle in the context of the great *Táin Bó Cúailnge* 'the cattle-raid

[1] `https://celt.ucc.ie//published/G301006/`

[2] Parts of the story are additionally found in a glossary from a 1700 manuscript.

[3] The personal name might also be related to *froech* 'fury'. Moreover, in a review of Meid's first edition (1968), Ní Chatháin (1969–1970: 75) suggests a connection with the Gaulish divine name *Vroicis* (dat. pl. in Latin), corresponding to Old Irish *froích* 'heather'. In other words, invoking a relation with a place-name may not be strictly necessary.

of Cooley'. Meid (1974) states that the original romance story ending in Fróech's death was reworked (Fróech was made to survive) to constitute a prequel (*remscél*) to *Táin Bó Cúailnge*.

The manuscript evidence has led Meid (1974: xxv) to believe that 'the archetype, apart from some corruptions and very few Mid. Ir. forms, has faithfully preserved the text and the language of an O. Ir. original', pointing to a composition date of around 750 or even 700 A.D. The editorial remarks include the important point that 'Middle Irish forms and spellings have been allowed to stand, even where the correct Old Irish forms could have been supplied from other manuscripts' (Meid 1974: xxv-xxvi).

5.2.3 Verb count

Text statistics are found in Table 5.1. This work deals primarily with Old Irish. Univerbated compound verb forms with absolute endings have not yet been integrated in the FST; for testing purposes these are therefore excluded in the weak verb count. An example of such a simple verb is prs. ind. 3sg. *fácbaid* 'leaves' < OIr. *fo·ácaib* (under this lemma in Meid 1974). Another example is prs. ind. 3sg. *oslaigid* < OIr. *as·oilgi* 'opens' and *ron oslaicis*, listed under the lemma *oslaigid* in Meid (1974) (but under Old Irish *as·oilgi* in the glossary in Meid et al. (2015)). Such forms could be encoded with a Middle Irish upper-level tag of the type `+MID_IR`. Finite-state methods dealing with non-standard language forms are discussed in section 6.4.5.

The low proportion of W1 and W2a verbs (section 2.4.4) in relation to the total amount of verbs, especially in terms of tokens (constituting less than a tenth of the overall number of inflected verbs), was not expected. Ignoring instances of the copula, substantive verb and defective verb form *ol* 'said' (section 5.5.2), which are the most frequently occurring verbs in the text, most inflected forms—based on a cursory inspection—belong to the various strong verb types (including those with suppletive stems, e.g., S1a *téit* 'goes', *luid* 'went').

Table 5.1 – Text statistics for *Táin Bó Fraích*.

		Lemma	Inflected form (tokens)
Verbs (excl. verb. nouns)	W1	16	31
	W2a	11	23
	Other verbs	101	598
	Total	128	652
Other POS (incl. verb. nouns)		660	2981
Grand total		788	3633

5.3 Small modifications to the transducer

5.3.1 Preverbs and infixed pronouns

After adding the compounds *imm·múcha* ‘suffocates’ and *ar·peitti* ‘plays music, entertains’ with the (pretonic) preverbs *imm·* and *ar·*, respectively, it was found that some changes were necessary to Code Example 4.7, to correctly arrive at forms such as *Immus·múchat*[4] ‘suffocate one another’ and *arus·peittet* ‘entertained them’, both found in TBF. It is more economical to derive the preverb with infixed pronouns from *immu-*, *immi-*, *ara*, *ari*, etc. than from surface forms without infixed pronouns where the historical vowel does not (generally) surface. However, the preverb with a vowel causes an undesired vowel coalescence with pron. 3sg. masc./neut. class A *-a*, as in *immu-a*. Rule two in Code Example 5.1 specifies a more general deletion rule for *-a* compared to its ‘predecessor’ (including switching rule one and two around so that the rule condition for *ní*, after which *-a* should be deleted, applies first).

Code Example 5.1 – Redefined rule part of `proclitic.script` for vowel coalescence with infixed pronoun 3sg. masc./neut. class A (section C.3.9 on page 213).

```
define vowCoalesc [
["^PRONa" -> 0 || {ní} _ ] .o.
[Vow -> 0 ||  _ "^PRONa" ] .o.
["^PRONa" -> a]
] ;
```

5.3.2 Delenition of θ

The passive preterite stem consonant θ becomes *t* after *s* in verb forms such as *briste*, from *brissid*, one of my test set verbs. After incorporating the stem entry for the W1 simplex *gataid* ‘takes off, steals’, it was found that delenition should also apply to *t* preceded by the preterite passive stem consonant, either resulting in *t* or *tt* (preterite passive plural augmented[5] *ro gata* as well as *ro gatta* ‘were taken off/stolen’ found in our text). Delenition issues were additionally found with prs. ind. pass. 3sg. *Fodáilter* of W2a *fo·dáili* ‘divides, distributes’ and *Ráite* of W2a *ráidid* ‘tells’. The relatively straightforward additional/amended replace rules in Code Example 5.2, based on Thurneysen (1946: §§ 137–141), solve these issues. Note, however, that these rules only cover the phonological processes in our text. There are more delenition processes, which are not catered for in the present implementation, e.g., *ro·ráitsem* (Thurneysen 1946: § 139), ‘we have said’, where /ð/ and preterite stem consonant *s* cause delenition of /ð/ to *t*, but probably realised as /d/.

[4] The preverb *imm-* can be understood as a preverb denoting mutuality (Thurneysen 1946: § 841), prefixed to the simple verb *múchaid* ‘suffocates’. I have taken this as a compound verb based on Meid (1974), who has listed this form separately in the vocabulary (under *i*). eDIL also has a separate headword *imm-múcha*, cf. `http://dil.ie/27885`.

[5] Meid (1974: 60) and Meid et al. (2015: 285) speak of a ‘perfect’ here.

Code Example 5.2 – Missing delenition rule now part of `13_se_phon_orth_cons.rule` (section C.2.13 on page 203).

```
[ θ -> t || d|l|n|s|t _ ] .o.
[ d -> 0 , t (->) 0 || _ t  ] .o.
```

5.3.3 Updated stems and an old inflectional ending

A form that wrongly turned out to be subject to syncope is ipf. 3sg. *timchellad* of *do·imchella* 'goes around'.[6] The FST initially produced *timmchled* (stem entry `ti^Mchellā`). Upon inspection of the stem entries it was found that this monolithic stem does not correctly represent underlying *to-imbe-cell* for the purpose of correct syncopation of the second syllable, although eDIL, somewhat unhelpfully, gives underlying *to-imb-cell*[7]. The corrected monolithic stem entries in `lexc` format, shown in Code Example 5.3 (`^M` is rewritten as either `m` or `mm` in this case, cf. `proclitic.script`, section C.3.9 on page 213). It should be noted that compound stem entries are by definition monolithic stems (section 2.4.1 and section 4.6.1), not derived from prehistoric forms, so I did not change a rule here, but just amended the stem entries for this particular lemma on an ad-hoc basis.

Code Example 5.3 – Updated (monolithic) stem entries for the compound verb *do·imchella* as part of `compoundW1.txt` (section C.4.1 on page 222).

```
@R.PV.TO@+imbi+PV2+cell+VROOT:i^Məchell
        W1;
@D.PV@+to+PV1+imbi+PV2+cell+VROOT:ti^Məchell
        W1;
```

Prt. 3sg. rel. *arabeiti*,[8] *arabeitte* remind us of the older prt. 3sg. *-i* ending with W2 verbs, which was dropped since it was undistinguishable from the prs. ind. 3sg. conj. (Stifter 2006: 200). Amending `se.lexc` just entails one extra line of code for this ending, so this ending variation was incorporated into the FST.

5.4 Pre-processing and `flookup`

The text was downloaded from CELT[9] and converted to `.txt` in `UTF-8` encoding. The bibliographic information as well as page and line numbers were taken out and to create a 'clean'

[6] In the most recent edition, this form has been amended to 3pl. *timchellat*, which makes more sense in the context (*Timchellat a tech* 'they make a circuit around the house'), conjecturing that the 3sg. *-d* ending found in the manuscripts might be an error already present in the archetype (Meid et al. 2015: 157).

[7] `http://dil.ie/17856`.

[8] The majority of manuscripts have *arbíth* (or *arbith*) from *ar·tá* 'is present, is in store' instead. According to Ní Chatháin (1969–1970: 76), this replacement of the manuscript readings happens 'without too much justification'.

[9] `https://celt.ucc.ie//published/G301006/index.html`

text file. It is worth pointing out that the CELT edition reflects the original editorial policy by Meid (1974) regarding word separation. This means that conjunct and verbal particles (e.g., *ní*, *ro* and *no*) are separated from the rest of the verbal complex by a space, while an independent compound verb (e.g., *Dolléici*) is presented as a consecutive string. Hyphens are mainly employed to mark nasalisation before a vowel (*n*-). This practice, as well as many other possible word and morpheme segmentation practices, were anticipated. Therefore, no further pre-processing was necessary.

The whole text was fed in. The utility `lookup` (Beesley & Karttunen 2003: 431–438), or `flookup`,[10] accompanying `foma`, is a runtime program invoked from the command line that applies a pre-compiled transducer to a word list, which may be extracted from a corpus. The output is a vertical list of (typically) the surface form followed by the morphological analysis (tags), separated by a tab. If the surface (e.g., orthographical) form has more than one parse, it is listed again, i.e., there are as many lines as there are possible morphological parses. Strings that are not part of the language of the transducer (not recognised) are accompanied by `+?`. Extracting unique words (types) from a corpus and testing the lexical transducer against a list of words typically involves `unix` commands, as illustrated in Code Example 5.4 and Code Example 5.5 (comments preceded by `#`), based on Beesley & Karttunen (2003: 332). For the creation of the final all-inclusive FST cf. `all.script` in Appendix C (section C.3.3 on page 210). The wordlist saved to `failures.txt` can be manually inspected and used to subsequently add new stem entries or improve the rule framework of the FST.

Code Example 5.4 – `unix` commands for creating a wordlist from a text.

```
# read in a corpus file
cat TBFcleanedup.txt | \
# tokenize, one word to a line (i.e., translate complement of
    specified characters into newline), save to file
tr -sc '[:alnum:]éíáóúÉÍÁÓÚ&-' '[\n*]' > wordlistTBF.txt | \
# read in file, sort alphabetically and filter types (uniq),
    save wordlist to file
cat wordlistTBF.txt | sort | uniq > TBFsortUnique.txt
```

Code Example 5.5 – Applying a wordlist to a lexical transducer and extract non-recognised strings.

```
# read in the wordlist
cat TBFsortUnique.txt | \
# apply the wordlist to the Old Irish lexical transducer
flookup oiAll.fst | \
# extract non-recognised strings (first field, the string
    itself, before '+?') and save to file
grep '+?' | gawk '{print $1}' > failures.txt
```

[10] `https://code.google.com/archive/p/foma/wikis/FlookupDocumentation.wiki`.

5.5 Results

5.5.1 W1 and W2a verbs

The results for the W1 and W2a verbs are in Table 5.2. For forms not recognised by my Old Irish morphological FST, I made a distinction between relatively trivial Old Irish orthographical variation on the one hand, and various miscellaneous issues—often less predictable in the context my rule framework—on the other. Before discussing the issues in section 5.6, I will give the recognition figures and overall statistics. The form *gataid* with an initial capital (equally prs. ind. 3sg.) has not been counted as a separate unique form, and neither has the already mentioned Middle Irish form *aisce*, for which cf. section 5.6.3.

Table 5.2 – Test results for weak verbs including lemmatisation. * = Middle Irish ending.

		Recognised by FST	**Not recognised by FST**		**Lemmatisation** (Dereza 2016)	
Lemma	Prs. stem type		OIr. spelling variation	Miscellaneous	Lemma	Correct lemma?
ad·ella	W1	*aidleth*				
anaid	W1	*Anait* (x2), *ansait*, *An*, *anat* (x2)		*Anfimni*	*ainimm*	No
caraid	W1	*Carthai*, *charus*, *chara*				
celebraid	W1			*Celebraid*	*celebraid*	Yes
do·alla	W1	*Dotallfasu*				
do·immchella	W1	*Timchellad*				
feraid	W1	*ferait*, *ferthair*, *ferais*				
fo·fera	W1		(1) *Fofirfe** (2) *Fonroíreth* (not implemented) (3) *foruíreth* (not implemented)		(1) *fo-fera* (2) *fo-fera* (3) *fo-fera*	(1) Yes (2) Yes (3) Yes
gataid	W1	*Gataid*, *gataid*, *gata*, *gatta*				
íadaid	W1	*n-íada*				
imm·múcha	W1	*Immusmúchat*				
lasaid	W1	*lastais*				
marbaid	W1	*marbam*				
múchaid	W1	*múchtha*				
rannaid	W1	*rannad*				
scaraid	W1	*Scarsat*				
ais(i)cid	W2a	*aisce*, *aisce**			*oc*	No
ar·áili	W2a			(1) *arrálad* (2) *arandálfarsa*	(1) *ard* (2) *arandálfarsa*	(1) No (2) No
ar·peitti	W2a	*aruspeittet*	(1) *arabeiti* (2) *arabeitte**		(1) *ar-peiti* (2) *ar-peiti*	(1) Yes (2) Yes
brissid	W2a	*brissis*				
do·léici	W2a	*Dolléici*, *Dolléicther*, *Dolléicetar*				
fo·dáili	W2a	*Fodáilter*, *Fodáile**	(1) *Fodlid*	(2) *fodailter*	(1) *fodlaide* (2) *fo-dáli*	(1) No (2) Yes
fo·ruimi	W2a			*forruma*	*fo-ruimi*	Yes
glúaisid	W2a	*glúaistis*				
imm·ráidi	W2a	*Immaroraid*				
léicid	W2a	*léicit*, *léicid*				
ráidid	W2a	*Ráite**	(1) *Ráidti* (2) *Rádid*		(1) *ráidid* (2) *ráidid*	Yes Yes

Table 5.3 – Results relative to unique inflected verb forms (excl. verbal nouns) contained in *Táin Bó Fraích*.

Verb category	Lemma (Meid 1974, OIr. only)	Unique inflected forms	Percentage out of total unique inflected forms	Recognised (correct morph. analysis)	Percent recognised within category	Percent recognised relative to total unique inflected verbs
W1	16	28	7.4%	23	82.1%	6.1%
W2a	11	22	5.8%	13	59.1%	3.4%
Subtotal Weak verbs (type 1 and 2a)	27	50	13.2%	36	72%	9.5%
atá (substantive verb)	1	29	7.6%	12	41.4%	3.2%
is (copula)	1	21	5.5%	15	71.4%	3.9%
ol ('said', defective verb)	1	1	0.3%	1	100%	0.3%
Other verbs	98	279	73.4%	0	0%	0%
Total	128	380		64		16.8%

The test results relative to all verbs in TBF are found in Table 5.3. The most important result is the number of correctly analysed (unique) inflected forms across the weak verbs category; this comes down to 36 recognised forms out of a total of 50 unique inflected forms (72%) across 27 genuine Old Irish W1 and W2a lemmas in the text. Furthermore, the totals row shows that 64 of the total of 380 unique inflected forms (16.8%) have been morphologically parsed in the text. The somewhat lower figures for W2a are mostly due to more inflectional and spelling variation, discussed in section 5.6.

Middle Irish endings sometimes coincide with an Old Irish one while constituting a different inflectional form (cf. section 5.6.3). For the W1/W2a verb forms that were recognised this is restricted to two occurrences of *·aisce* from W2a *ais(i)cid*, discussed in section 5.6.3. With the copula and substantive verb, which are not at the heart of my study, I did not fully investigate ambiguous inflection (homographs). One example is the occurrence of *is*, analysed as a non-relative prs. ind. 3sg., while in reality it used as a relative form in TBF (rather than appearing as Old Irish relative *as*).

5.5.2 Results across all words, and compared to other texts

I encoded some additional proper names and function words in the morphological FST. Table 5.4 displays results for all words, categorised by both type (unique forms) and tokens (instances of the same form). These scores are compared against four other texts, edited by Greene (1955), also generally dated to the Old Irish period: *Fingal Rónáin* (Table 5.5), *Orgain Denna Ríg* (Table 5.6), *Esnada tige Buchet* (Table 5.7) and *Orgguin trí mac Diarmata meic Cerbaill* (Table 5.8).[11] The average recognition score for types in these four additional texts is 10%, showing that the result for TBF (9.6%) is consistent across texts. In other words, seeing that I implemented only a small subset of verbs, this is a good result and strengthens the hypothesis that implementing further verbs is likely to show consequential gains. The somewhat higher score for tokens in TBF is largely attributable to the most frequent verb in this text, defective *ol* ‘said’ (102 instances out of 652 verb tokens in total). This verb form has been encoded in the FST but is very infrequent across the other four texts. Specific linguistic issues relative to TBF are discussed in section 5.6.

Dereza (2016) has already tested her Lemmatiser on TBF, reporting on a 81.07% recall score after improving the Lemmatiser based on a sample set of texts.[12] Although my score of around 10% is much lower than this figure, it should of course be borne in mind that I did not implement all word categories present in the text. However, 10% of words in the text receive a full morphological parse by the FST, not just a lemma. The gain of employing the Lemmatiser in relation to non-recognised verb forms is dealt with in section 5.7.

[11] All available in digital format on `https://celt.ucc.ie/`.

[12] For this test measurement cf. section 3.5.

Table 5.4 – Results across all words in *Táin Bó Fraích*.

	Types	Tokens
Analysis	155	1106
No analysis	1463	2527
Total	1618	3633
Percentage analysed	9.6%	30.4%

Table 5.5 – Morphological analysis results across all words in *Fingal Rónáin* (Greene 1955).

	Types	Tokens
Analysis	88	500
No analysis	949	1734
Total	1037	2234
Percentage analysed	8.5%	22.4%

Table 5.6 – Morphological analysis results across all words in *Orgain Denna Ríg* (Greene 1955).

	Types	Tokens
Analysis	66	310
No analysis	623	1019
Total	689	1329
Percentage analysed	9.6%	23.3%

Table 5.7 – Morphological analysis results across all words in *Esnada tige Buchet* (Greene 1955).

	Types	Tokens
Analysis	50	186
No analysis	383	547
Total	433	733
Percentage analysed	11.5%	25.4%

Table 5.8 – Morphological analysis results across all words in *Orgguin trí mac Diarmata meic Cerbaill* (Greene 1955).

	Types	Tokens
Analysis	82	386
No analysis	707	1074
Total	789	1460
Percentage analysed	10.4%	26.4%

5.6 Discussion

5.6.1 Ambiguity, segmentation, editorial policy

The task of a morphological FST is to present all the possibilities. Disambiguation is not part of its remit. Since not all word categories have been covered, a noun like *fer* 'man' is now analysed as a prt. 3sg. conj. of *feraid* 'pours', etc. More insightful examples in the context of this project, exhibiting the parsing power of the FST, include Middle Irish *ragatsa* from Old Irish suppletive fut. stem *rig*, *reg* of *téit* 'goes' (Schumacher 2004: 548), cf. (24). This form is from Old Irish fut. 1sg. *rega* with a petrified suffixed pronoun neuter *it* instead of *i*, used with 1sg. fut., 1pl. prs. and 1pl. fut. forms in Middle Irish (Breatnach 1977: 104).

As strong and suppletive stems are not covered in my implementation, the (unexpected) analysis accompanying this verb form is the one in (25).[13] This example illustrates that the FST can (and will) interpret proclitics and stems that are immediately consecutive, which in this case is perhaps too powerful an approach and causes unnecessary ambiguity.

(24) **rag-at=sa**
rag-SUFF=SUFF
go.FUT-1SG=EMPH.1SG
'*I* will go'

(25) **r-a-gat=sa**
ro-INFIX-steal=SUFF
AUG-PRON.3SG.N-steal.PRS.SUBJ.1SG=EMPH.1SG
'*I* may have stolen it'

Meid (1974) has in part already done the work for us: he aided the reader/learner by separating out conjunct particles and the augment *ro* from the stressed part of the verbal complex (deuterotonic compound verbs constitute one consecutive string). While the example *ragatsa* above might have benefited from a segmentation approach whereby proclitics like *ro* are not expected to be immediately consecutive to the stem (causing the unexpected analysis), one can imagine many situations where a verb form with an immediately consecutive proclitic *will* lead to a less ambiguous analysis by the FST, thanks to the built-in morphotactic restrictions which disambiguate non-possibilities (cf. sections 4.7 and 4.8).

An example of a situation where a verbal complex represented by a consecutive string would be beneficial in disambiguating terms is *ro charus*, augmented prt. 1sg. rel. 'whom/that I have loved', from *caraid* 'loves'. Meid (1974: 50) guides the reader in the glossary by specifying that this is a relative construction (the form in question is cited as *-charus*). The diagnostic of lenited *c*, as well as the prefix and stem being one word (*rocharus*), would have resulted in the FST recognising this consecutive string as a relative. Since *charus* is preceded by space, the FST now returns (ambiguous) tags for the proclitic and verb separately, as illustrated in (26) and (27), respectively.[14]

[13] `ro+AUG+PRON+A+3P+SG+NEUT+PROCL_JUNCT+LEN+gat+VROOT+W1+PRS+SUBJ+CONJ+1P+SG+EMPH+1P+SG.`

[14] Interestingly, if an online version of the most recent edition (Meid et al. 2015) had been available, the input would have been *ro·charus*, with the augment and verb being consecutive, avoiding the ambiguity discussed here (the presence of a mid-high dot as a separation marker of proclitic and tonic part of the verbal complex has been anticipated in the implementation, cf. section 4.8.1). Note further that a prs. 3sg. absolute relative is not given as an option by the transducer (although the *-us* / *-as* variation is trivial). A leniting relative is not encoded in this case as it is a development that becomes prominent only in the later Old Irish glosses compendia (Thurneysen 1946: 495(b)). While the current implementation aims to cover the full breadth Old Irish grammar, variation of this nature potentially deserves explicit upper-level tags in order to differentiate between normalised / canonical forms and various types of (later) variation, encoded in subsequent transducers (cf. section 6.4.5).

(26) a. `ro+AUG`

b. `ro+AUG+REL+LEN`

c. `ro+AUG+REL+NAS`

(27) `LEN+car+VROOT+W1+PRT+CONJ+1P+SG`

The issue here reflects the observations in section 4.4.2 on dependencies that might transcend the word boundary (i.e., a space), typographical variability and editorial policy. Filtering out consecutive non-possibilities is not part of the work described here. The string in (27), for example, could have been preceded by anything that causes *c* to be lenited, e.g., *not charus* 'I loved you'. In a subsequent step, a morphosyntactic disambiguation process would need to be invoked, in this case probably on the basis of the final `+LEN` tag as part of *ro* (relative), and the initial `+LEN` tag as part of *charus*. Such a morphosyntactic constraint grammar will also disambiguate between the grammatical parses relative to *fer*, which, as has been shown above, can be both a noun and a verb.[15] POS tagging, as well as systematically investigating inflectional syncretism (also seen with copula and substantive verb forms), is outside the scope of my work. However, section 5.6.3 will highlight Middle Irish 'false positives': forms with Middle Irish endings parsed as if they were Old Irish. These forms have also been explicitly marked in Table 5.2 (with a *), for both recognised and non-recognised forms.

5.6.2 Issues not covered in the FST modelling framework

A further issue was found with the form *timchellad* (mentioned in section 5.3) after subjecting the text to `flookup`. The form occurs with an upper-case letter (as it is sentence-initial), a feature which results in an independent imperative parse by the FST. However, it is not a 3sg. imperative here but a 3sg. ipf. This form assumes its prototonic base even though it is in independent position (and not imperative), a process that can be seen with compounds with *to*, *fo* and *ro* in case vowels meet a the clitic boundary (McCone 1997: 3), cf. also section 2.2.1. This issue can be solved by changing the conditions for upper-casing in the transducer discussed in section 4.10. However, this was felt too substantial a change to the FST as globally allowing capitalisation with prototonic bases would not only effect cases such as *do·imchella*, but any form of any compound verb, many of which are not expected to have prototonic inflection (univerbation) in independent position (apart from the imperative). In other words, the restrictions to upper-casing of word-initial symbols are in place to prevent prototonic yet apparently independent forms such as *Teilced* (deut. *do·léiced*) and *Aidled* (deut. *ad·ellad*) to be recognised as 3sg. ipf., past subj. and prt. pass. Alternatively, one can relax the capitalisation rules to cover all possible interpretations for, e.g., univerbated *Timchellad*, which would, however, entail—in the current implementation at least—the generation of the above-mentioned wrong or unexpected 3sg. analyses with all compound verbs (overgeneration).

[15] Such a disambiguation rule could involve specifying that a noun, as opposed to a verb, cannot be preceded by word classes such as conjunct particles, the augment and preverbs.

The W1 verb *celebraid* ‘bids farewell’, from Latin *celebrāre*, is subject to syncope in my implementation, leading to prs. ind. 3sg. *ceilbrid.* I do not see any reason to change my syncopation rules based on this form as the consonant sequence *-lbr-* is phonotactically possible. The form almost certainly occurs without syncope in Old Irish due to the (known) resemblance to its Latin cognate, from which it was borrowed. Furthermore, eDIL notes ‘a preponderance of *cel-* over *ceil-* perh. due to Lat. infl’.[16] In other words, this form is subject to analogy, which is the reason for the absence of syncope (cf. section 4.6.4.4 for a discussion of analogy and syncope). I will return to problems with the application of syncope below.

The inflected forms of the W2a verb *ar·áili* ‘arranges’ in TBF show a couple of problems. A fut. 1sg. form *arandálfarsa* with a deponent ending (*-ar*) occurs. Deponent endings have not been catered for in the FST. Generating the fut. 1sg. form using the FST (ignoring the infixed pronouns) gives us *ar·áiliub*, a variant manuscript reading and the expected form (*arandailiubsa*, with infixed pronoun 3sg. neut. and emph. particle 1sg.; Meid et al. (2015: 209)). The second occurrence of this verb is *arrálad* (*arr·álad*), a rel. prt. pass. 3sg. form that seems to realise the nasalisation on the final consonant of the preverb (*-rr*), rather than on *ál-*. For an Old Irish relative one might expect the preverb to appear as *ara-*. However, in the most recent edition, this form has been analysed as perf. pass. 3sg. rel. *ar·rálad* (Meid et al. 2015: 238, 275), with infixed and stressed *ro*. This analysis did not come up as a possible interpretation either as I did not implement the augmented monolithic stem `rāl`.

Two further issues refrain these inflected forms from being covered by the language of the FST to begin with. Firstly, the 3sg. pron. masc./neut. Class C, due to its many surface shapes (*id*, *did*, *d*)[17] has not been integrated in `proclitic.lexc` yet.[18] The allomorphic variation probably justifies the creation of a full-form lexicon for each preverb with this specific infix. Secondly, the non-palatal consonant before the endings in these forms hides the fact that one is dealing with a W2a verb. As mentioned above, the fut. 1sg. form generated by the FST (ignoring the infixed pronoun) is *ar·áiliub*, while a 3sg. pret. pass. (analogous to the interpretation *arr·álad* per Meid (1974)) maps to *ar·áiled* and *ara·n-áiled* (rel.). These forms are thus contained in the FST with a palatal stem-final consonant.

Although strictly speaking not part of the category of verbs focused on, I tested the implementation of W2b verb *con·tuili* ‘sleeps’, which is represented in TBF by dependent/prototonic prs. subj. 3pl. *·comtalat.* First of all, the root vowel is (wrongly) syncopated due to my FST rule framework,[19] resulting in `comtlet`. In Code Example 4.16, under section 4.6.4, I have provided code for dealing with phonotactically impossible consonant clusters arising from syncope. The (wrong) application of syncope resulting in `comtlet` might be circumvented by changing this rule. However, the solutions in the rule framework to counteract syncope are of a tentative/ad-hoc nature and need revision in a subsequent stage of my project regardless in order to deal with a broader range of cases that do not show syncope (whether represent-

[16] `http://dil.ie/8552`.

[17] The infixed pronoun in *arandálfarsa* (*ara-n-d·álfarsa*) is neut. *-d* after the nasalising relative marker *-n-*.

[18] For the current version of the file cf. section C.1.2 on page 179.

[19] This was tested out by inputting the stem entry `comtəlī`.

ing irregular syncope or expected exceptions to a generally applied rule, e.g., phonotactically impossible consonant clusters).

More importantly, `ī` is lowered in my approach before a non-palatal consonant (prs. subj. 3pl. `-t`). Rules of this type are based on the test set of forms including prs./subj. 3pl. *·léicet* (`léicī-t`) and *·reilcet* (`reiləcī-t`). McCone (1997: 27–28) points out that the present stem-final consonant with W2 verbs is often not palatalised when there is no syncope of (what I have encoded as the stem vowel) `ī`, hence, for example, W2a 3sg. *fo·rrumai*[20] 'puts' and *rádas* (instead of *ráides*) 'which he speaks'. Under the headword for the latter form, i.e., *ráidid*, eDIL states 'in O.Ir. occas. treated as ā-st',[21] which explains the non-palatal stem-final consonant.

I was not familiar with this fluctuation of stem-final consonant quality in W2 verbs and hence did not implement this variation. The problem is undoubtedly solvable by reformulating the way W2(a) stem entries are formulated or changing the inflectional rules; this, however, proved to be not feasible due to time restrictions and the fact that these complexities were not anticipated when laying down the finite-state rule framework.

The W2a verb *fo·dáili* 'divides, distributes' is potentially subject to the same problem as outlined above, except that no 'problematic' inflected forms with variation in stem-final consonant quality occur with this verb in the text. The inflected forms that do occur show a different set of issues (apart from a small issue with delenition with the prs. ind. pass. 3sg. *Fodáilter*, which has already been discussed in section 5.3). One of the problems reflects a relatively minor issue related to spelling variation: imp. 2pl. *Fodlid* for *Fodlaid* as produced by the FST—less ambiguous in terms of the quality of the consonant cluster *-dl-*. Spelling variation is more substantially covered in section 5.6.4.

The third inflected form of *fo·dáili* occurring in TBF is dependent subj. pass. 3sg. *·fodailter*. This inflection is contained in the transducer in the shape `fodlaither` as a result of (expected) syncope of the second (rather than the third) syllable. Discussing syncopation of vowels in third syllables, Ó Crualaoich (1999: 95) notes that 'there was a tendency in Old Irish to spread unsyncopated stem plus syncopated following syllable beyond their original range'. The latter provides the example *fodlad*, imp. 3sg. of *fo·dáili*, which he contrasts with non-syncopated *todālib*, *todāilib* from the Milan Glosses (verbal noun dat. pl. < *do·dáili* 'pours out'), showing that syncopated and non-syncopated forms in compounds with the same verb root may coexist.

The inflected form *Anfimni* of simple *anaid* 'stays, stops' is a fut. 1pl. abs. based on the position in the sentence. The transducer produces *ainfimmi* and *anfaimmi* instead. According to Meid et al. (2015: 192), this form 'is for *anfimmi-ni*, with ending shortened before the suffixed personal pronoun (*sic.*) *-ni* (older *-sni*)'.[22] I might add that the 'shortened' form *anfimni* is suspiciously close to *anfimmi* without the suffixed emphasising particle.

[20] Accordingly 3sg. prs. subj. rel. *forruma*, found in TBF, rendered as *forruimea* instead by the FST (nasalising relative is correctly incorporated). Note that Meid (1974), Meid et al. (2015) and eDIL (`http://dil.ie/24043`) have the lemma *fo·ruimi* instead.

[21] `http://dil.ie/34742`.

[22] It should be noted that *anfimmi-ni* would not have been recognised either, as the vowels surrounding *-nf-* are ambiguous as to consonant quality of this consonant cluster (more on which in section 5.6.4). The transducer currently only generates normalised forms with consonant quality being unambiguously encoded in the spelling.

5.6.3 Middle Irish forms

The closest I get to fut. 3sg. *Fofirfe* of *fo·fera* ‘causes’, according to Meid (1974: 59), is *Fofeirfea*.[23] The form found in the text employs the ending *-e* instead of *-ea* for the 3sg. fut conj., which is typical for Middle Irish (Breatnach 1994: 316). This ending variation is due to unstressed final vowels falling together as ə (Russell 2006: 990). Similarly prs. subj. 3sg. *·aisce* for Old Irish *·aiscea* (< *ais(i)cid* ‘returns’, etc.), prt. 3sg. rel. *arabeitte* for Old Irish *arabeit(t)i*, *arapeit(t)i* (< *ar·peiti* ‘plays music, entertains’), *ráite* (occurring with a word-initial capital in our text) for Old Irish *ráit(t)i* (cf. section 5.3.2), and prs. ind. 3sg. *Fodáile* for Old Irish *fo·dáili* (< *fo·dáili* ‘divides, distributes’).

While *aisce*, *Ráite* and *Fodáile* are recognised due to the fact that the Middle Irish endings inadvertently coincide with (other) endings relative to a more narrowly defined Old Irish paradigm, *Fofirfe* and *arabeitte* additionally show orthographical variation in their stem (relative to normalised Old Irish). Morphological parsing with these two specific forms does not lead to ambiguous results, but is entirely unsuccessful in the first place. Therefore, rather than classifying their non-recognition as (additional) Middle Irish grammatical variation, they are treated as deviating from a (superficial) orthographical norm, i.e., Old Irish spelling variation, as can be seen in Table 5.2. Spelling variation found with these and other forms is discussed in the next subsection.

5.6.4 Normalised Old Irish and spelling variation

Catering for minor Old Irish spelling variation with W1 and W2a verbs would have boosted the recognition rates further. Admittedly, the delenition rules and the encoding of some of the resulting variation (e.g. *gata*, *gatta*), discussed in section 5.3.2, already cater for what can be termed spelling variation. One could go even further and encode spellings of the type *Ráidti*, underlying *rádith-i*, prs. ind. 3sg. of *ráidid* ‘tells’, with suffixed pron. 3sg. neut, i.e., ‘tells it’. This form is subject to syncope, resulting in /ð′θ′/, which is delenited to [t′]. The sequence <dt> employed in this form, for [t], is therefore an etymological spelling.[24] This form has been normalised to *ráitti* in the recent edition of TBF by Meid et al. (2015: 148), and the editors did the same with *ráite*, already mentioned in sections 5.3.2 and 5.6.3.

Not fully implementing all possible spelling variation was a deliberate choice; it is envisaged that—rather than working on an ad-hoc basis—a separate module (which could be an FST) should augment the ‘clean’ FST that systematically rewrites normalised spellings into possible spellings. This aspect is further discussed in section 6.4.5. A different and probably more elegant and economical approach, already referred to in section 4.3.4 and section 4.6.4.3, is to operate with underlying symbols such as phonemes, which can be subsequently mapped,

[23] Due to time constraints, I have not managed to investigate a meaningful way of implementing the stem-vowel variation seen with independent and dependent augmented (perfect) 3sg. passive forms *Fonroíreth* and (·)*foruíreth* of the same lemma. Meid et al. (2015: 258) give the historical derivation **fo-ro-ḟerath* and use the spelling *oí* for both forms.

[24] Compare *·midter* (/m′ið′/) ‘thou judgest’ (Thurneysen 1946: § 137).

by a separate transducer, to all possible graphematic variants. This would involve a non-trivial (yet potentially meaningful) rewriting of the code, however.

An example of a variant spelling is *b* for *p*, which 'in initial position is not clearly distinguished from *b*-' (Thurneysen 1946: § 920). An example of this variation is found in TBF: *arabeitte* and *arabeiti*[25] from *ar·peitti*[26] 'plays music, entertains'. In discussing relative *ara·beiti*, Meid et al. (2015: 175) observe that:

> The pronunciation *b* would be understandable here as mutation of *p* in a nazalising relative clause, but this only coincidental, since *p* was not yet firmly integrated into the system of mutations and was only partially affected by lenition, but hardly by nasalization.

Some variant readings in the manuscripts have *p*- here. It must be stated, though, that the spelling *b* is common in Middle Irish for the voiced allophone of /p/ (Breatnach 1994: 228).

Much of the spelling variation centers around the orthographical marking of palatal and non-palatal consonants. In my implementation, I adhere as much as possible to normalised or canonical spelling, approximating Classical Old Irish (following the likes of Stifter (2006)). Listed below are W1/W2a verb forms not recognised during testing, solely due to the absence of <e>, <i> or <a>.

1. *fo·firfe*, future 3sg. (with Middle Irish ending) of W1 *fo·fera* 'causes'; normalised *fo·feirfe* (or, rather, 3sg. 'genuine' Old Irish *fo·feirfea*).

2. *fodlid* (occurring with a word-initial capital in our text), imp. 2pl. of W2a *fo·dáili* 'distributes'; normalised *fodlaid*.

3. *rádid* (occurring with a word-initial capital in our text), prs. ind. 3sg. of W2a *ráidid*, 'tells'; normalised *ráidid*.

It should be noted that much of this variation is present across linguistic resources for Old Irish. For example, both Meid (1974) and Meid et al. (2015) list the lemmas *léicid* and *do·léici*, while eDIL has *do-léci* for the latter. Strachan (1949) uses the spelling *berid* for *beirid* (eDIL, etc.), but does include the *i* in *léicid* (rather than *lécid*) to mark that the stem-final consonant is palatal. The difficulty, obviously, is that *i* is not always shown in texts.

5.7 Lemmatisation

The weak verb inflections in TBF not recognised by (i.e., not contained in) my lexical transducer were subjected the Early Irish Lemmatiser (Dereza 2016) based on eDIL (described in section 3.6.1). The results are shown in Table 5.2 above on page 111, including a complete

[25] But cf. footnote 8.

[26] Note verb root *sēt*, cf. `http://dil.ie/4244`.

list of inflected forms already successfully morphologically parsed by the FST. An additional 10 inflected verb forms belonging to the present stem classes W1 and W2a, which were not identified by the FST, now receive a lemma. This results in a score, including lemmatisation, of 46 unique[27] inflected W1/W2a verb forms (36 recognised by the transducer, cf. Table 5.3) out of 50 unique W1/W2a verb forms in total (92%) for this particular text.

The various non-recognised forms have been discussed in the subsections in section 5.6. Not all of these reflect orthographical variation in Old Irish. Forms such as *fodailter* and *Celebraid*, found under 'Miscellaneous' in Table 5.2, adhere to Old Irish spelling norms but diverge from the expected syncope pattern and are thus more accurately described as Old Irish grammatical variants. Obviously, the meaning of 'grammatical variant' here is relative to predictable processes and mechanically implemented rules, generating expected or regular forms. Imp. 2pl. *fodlid*, from *fo·dáili* 'divides, distributes', is lemmatised as the adjective *fodlaide*, contained in eDIL,[28] which specifies that *fodlaide* is a participle of *fo·dáili*. Although lemmatisation is inaccurate (*fodlid* cannot be an inflected form of the *io-iā* adjective *fodlaide*), a reader/user ending up looking up *fodlaide* in eDIL at least will be informed that this adjective/participle is from the verb *fo·dáili*.

5.8 Synthesis

This chapter has discussed a case study that involved testing the Old Irish Finite-State Transducer on the Early Irish text *Táin Bó Fraích* (TBF) (Meid 1974, Meid et al. 2015). The purpose of the case study was to establish the extent to which the developed FST, adhering to approximate, normative Old Irish, is successful in dealing with a text that shows various types of linguistic variation. The share of Middle Irish forms under W1/W2a lemmas turned out not to be very extensive, and 'successful' recognition is in some cases due to Middle Irish endings coinciding with (different) Old Irish ones (e.g., *·aisce*). Section 5.6.4 has discussed typical spelling variation such as the (non-)marking of palatal and non-palatal consonants.

The recognition scores for Old Irish W1 and W2a by the FST were found to be 36 out of 50 unique inflected forms in TBF (72%). An additional 10 (unique) inflected forms are recognised by the Early Irish Lemmatiser (Dereza 2016) leading to a recognition score of 46 out of 50 unique inflected forms (92%) in this specific text (section 5.7). Together with the other verb types covered (substantive verb, copula, *ol* 'said'), 16.8% of verb forms (types) were recognised when set against all verb forms in the text (Table 5.3). After the incorporation of function words and a selection of personal names, 9.6% of word types in the text have been covered in terms of morphological analysis.

The test results reported on in other projects dealing with historical languages, discussed in section 3.6, mainly reflect lemmatisation and POS tagging efforts for historical texts, often

[27] Technically 46, as the form *·aisce* represents two different inflected forms: an Old Irish prs. subj. 2sg. and a Middle Irish prs. subj. 3sg. (both found in TBF).

[28] `http://dil.ie/22619`.

using already existing resources. The present work deals with a new system and is mainly confined to morphological parsing. The literature seems to suggest that few rule-based morphological analysers exist specifically built for a historical language, with the exception of parsers for ancestral languages such as Sanskrit and Ancient Greek. However, performance of these systems relative to texts is not documented. The recognition score obtained by my system is therefore hard to compare to other projects dealing with morphological parsing for historical texts. Moreover, my project is mainly concerned with verbs. However, the fact that a recognition score of around 10% is consistently found across four other Old Irish narrative texts suggests that the morphological FST is not just tailored to TBF, but will reach similar scores for other Early Irish texts.

Chapter 6

Conclusions and future work

6.1 Introduction

This chapter concludes the current work. It recapitulates the challenging backdrop of the lack of resources (section 6.2) and defines the contribution to scholarship as part of this thesis: a morphological parser for Old Irish verbs (section 6.3). Section 6.4 explores the envisaged bidirectional framework for linking Early and Modern Irish verb forms, building on Figure 3.3 on page 52. The most fundamental mapping strategy is to interconnect the Finite-State Transducers (FSTs) for Old and Modern Irish by creating a mapper between upper-level (lexical-level) strings. This method is described in section 6.4.2, using a selection of verb forms from the text *Táin Bó Fraích* as an example. Linking verb forms using lemmatisation is the subject of section 6.4.3. Section 6.4.4 discusses normalisation methods by employing Dereza's (2016) Early Irish Lemmatiser. Creating separate FSTs to deal with variants and unknown forms is the subject of section 6.4.5. The concluding remarks in section 6.5 justify, as well as reflect on, the deliverables relative to the present work and discuss some long-term challenges and opportunities. A synthesis follows in section 6.6.

6.2 Past and present: bridging the gap

The impetus of this work was the observation that there is currently insufficient digital support to systematically and comprehensively identify cognate verb forms across the historical periods of Irish. This impedes a systematic diachronic investigation of the Irish verbal system using computational means, the broader aim that underlies this thesis. Appendix A was carried out to establish the exact nature of the hiatus in digital support. In section 1.5 I have summarised this hiatus in terms of a 'lexicographical gap'. Projects whose aim was similar to the one in this thesis are either dormant, unavailable or unfinished; this is a clear indication of the challenges surrounding the computational exercise of automatically linking cognates.

The focus on (Classical) Old Irish in this project has been justified throughout this thesis: it is both a normative and reasonably homogeneous language stage, and it is well resourced, particularly in terms of printed works. At the same time, however, and in contrast to Modern Irish, little effort has been paid to NLP for Old Irish, and morphological parsing initiatives are virtually absent. This situation is undoubtedly at least in part due to the complexity of Old Irish grammar, caused by an often non-trivial interplay between morphology and phonology, especially with regard to verbs, as discussed in Chapter 2. Another reason for the lack of automatic parsing methods for Old Irish is that no 'off-the-shelf' language-independent algorithm or methodology is available for historical text processing; computational solutions for dealing with historical texts are mostly geared towards individual languages and based on language-specific digital tools and resources available for them, as illustrated by the projects surveyed in section 3.6.

The creation of a tool generating full paradigms for normalised Old Irish verbs was deemed necessary for comprehensively bridging the Old and Modern Irish period. In Figure 3.3 I have illustrated the philosophy of a 'two-pronged attack', with an Old and Modern Irish morphosyntactic tagger as anchor points at the opposite ends of the chronological spectrum, accompanied by standardisation methods orientated towards either Old Irish (for non-normative and Middle Irish forms) or Modern Irish (for early modern and pre-standard forms). In section 6.4 I will propose a roadmap as part of future work relative to my project.

6.3 Contribution of this thesis: a morphological parser for Old Irish verbs

The creation of a morphological analyser (and generator) for Old Irish verbs represents the most substantial and innovative part in the current work, as well as being one of the most novel approaches in the wider discipline of NLP for Early Irish. Chapter 4 has shown, richly illustrated with Code Examples, how the Old Irish verbal system can be successfully captured by the computational paradigm of finite-state morphology using the finite-state toolkit `foma` (Hulden 2009). Coming up with a computationally workable definition of stem is by far the most significant achievement in this thesis. I have called this unit a monolithic stem (sections 2.4.1, 4.6.1 and 4.6.2): a non-derived multi-morpheme base consisting of the string containing everything from the stressed element of the verbal complex up until the ending. This 'middle part' of the verb is most liable to variation, ultimately due to the stress system of Old Irish. The alternative approach, deriving stems by formulating morphophonemic rules applying to underlying roots, was not deemed feasible due to the sheer inflectional variation seen with, for example, deuterotonic vs. prototonic bases. The finite-state morphology paradigm was found to suit the workings of the Early Irish verbal system well; having two levels at one's disposal facilitates a straightforward and informative mapping between an underlying and surface string that often bear no immediately apparent relationship to each other.

This work has focused on weak verb types W1 and W2a due to predictable stem and ending formation. However, my aim from the beginning has been to facilitate the incorporation of compound verbs and strong verbs, which pose the biggest linguistic challenges, into the morphological FST. As Chapter 4 has shown, the current finite-state concatenation infrastructure allows relatively easy incorporation of any simple, compound, weak or strong verb. Proclitics, inflectional endings and other suffixes can be conveniently defined with `lexc` continuation classes. For those verbs that have been implemented, my current system already greatly surpasses the amount of inflected verb forms documented in any text edition, dictionary or grammar. Moreover, these forms incorporate (morphotactically legal) consecutive prefixes and suffixes.

As part of this thesis I set out to explore the balance between automatic and manual methods. The monolithic stem is crucial in this regard; without these stems, inflectional rules would be very hard to define. Identifying the monolithic stem(s) for a verb is the most knowledge-intense aspect of the implementation as it demands a thorough insight into Old Irish verb morphology. Historical derivation of stems from roots is not part of my approach; my framework relies heavily on pre-defined stems. A cursory exploration of sources (section 2.4.3) has revealed that there are no exhaustive lists of Old Irish verb roots and the preverbs that they combine with, let alone the stems that one has to operate with. Furthermore, analogy interferes with the generalisability of rules operating across the lexicon and the inflectional endings. Analogy has been discussed in relation to syncope in section 4.6.4.4. Academic collaborations with specialists in the field, especially to arrive at the monolithic stems for each verb, is essential in terms of future expansion of my project and establishing the exact nature of the above-mentioned balance. I will return to research prospects in the next section.

Due to the lack of exhaustive lists of roots, verbs, and verb types in Early Irish, it is not easy to put a finger on the exact share of verb forms and inflectional variation that has been covered by my computational system. However, a score of 72% reflecting correctly morphologically parsed Old Irish W1 and W2a verbs in *Táin Bó Fraích* (both simple and compound) is an important and promising empirical finding. Moreover, looking at all unique words in the text (types), a consistent recognition score of around 10% can be observed across five narrative texts after incorporating a few other frequent verbs, function words and personal names.

Building a morphological FST for Old Irish verbs has proven to demand an in-depth knowledge of Old Irish grammar; translating the complexities into a language that a computer can understand was found to be a far from trivial exercise. The ‘computational journey’ is well described in Beesley & Karttunen (2003: 287), who point out that:

> Formalizing your models [...] will inevitably highlight possibilities and gaps that you didn’t imagine; and even the printed description of your language will soon prove to be inaccurate and incomplete, intended as informal guidance to thinking humans rather than formal descriptions for a computer program [...] Building and testing a morphological analyzer can therefore be an important part of the linguistic investigation itself.

My contribution to a computational and more formal approach to Old Irish grammar is the most important research outcome of this thesis. At the same time, it has resulted in a shift of focus on Early Irish rather than on Modern Irish in the course of my project. Investigations into ways of mapping between Old and Modern Irish cognate verb forms—the original impetus for the work—has consequently become a subsidiary aim of the thesis. The next section explores future directions of my project.

6.4 The future: a roadmap for mapping verbal cognates

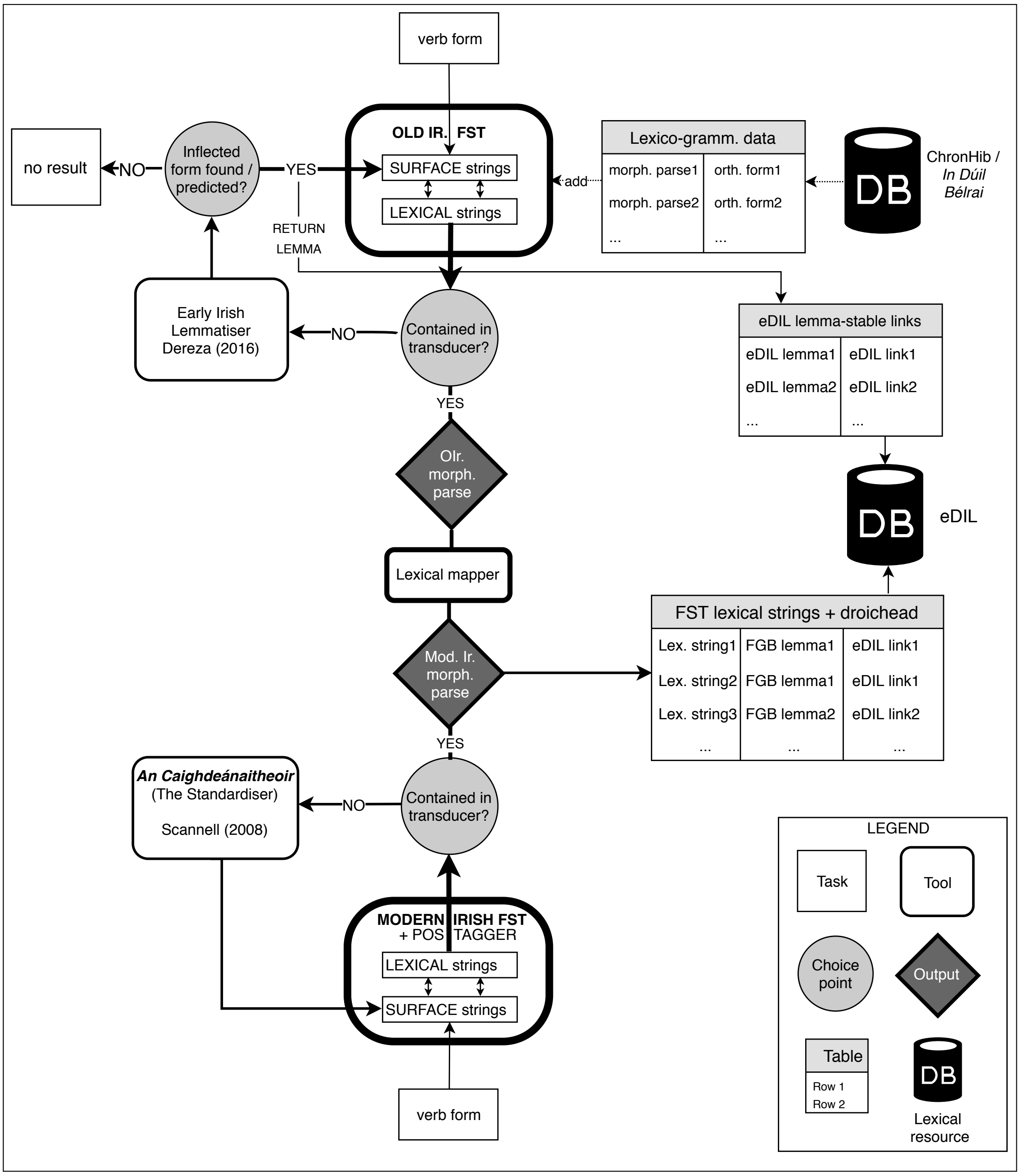
6.4.1 Two mapping methods

The diagram in Figure 6.1 shows the mapping and linking framework based on the 'two-pronged attack' discussed in section 3.7. Vertically the diagram shows the diachronic level: how to go from Old to Modern Irish verb forms, and vice versa. Two mapping methods are proposed:

1. Mappings between the lexical level of my Old Irish FST and the Modern Irish FST (Uí Dhonnchadha & van Genabith 2006), cf. section 6.4.2.
2. Lemmatisation using Dereza (2016) and a table incorporating `droichead` (Scannell 2018), a list of mappings between entries in the modern dictionary *Foclóir Gaeilge-Béarla* (*FGB*) and eDIL; cf. section 6.4.3.

6.4.2 Lexical-level mappings using two Finite-State Transducers

This method entails creating mappings using the lexical levels of the morphological FSTs for Old Irish (Chapter 4) and for Modern Irish (Uí Dhonnchadha & van Genabith 2006), respectively. In order to facilitate these mappings, a list of tag mappings needs to be specified. A significant challenge, albeit a very interesting linguistic one, is to encode mappings between stems surviving in Modern Irish and their often abstract historical roots in Old Irish (if a verb root, optionally with what used to be lexical preverbs, survives as a Modern Irish stem, of course). As part of this mapping method, detailed grammatical information (tags) is retained (alongside root/lemma).



Lexico-gramm. data	
morph. parse1	orth. form1
morph. parse2	orth. form2
...	...

eDIL lemma-stable links	
eDIL lemma1	eDIL link1
eDIL lemma2	eDIL link2
...	...

FST lexical strings + droichead		
Lex. string1	FGB lemma1	eDIL link1
Lex. string2	FGB lemma1	eDIL link1
Lex. string3	FGB lemma2	eDIL link2
...	...	...

Figure 6.1 – Diagram showing diachronic mappings and linking of historical Irish cognate verb forms.

Table 6.1 – Lexical-level mappings: a selection of verb forms in *Táin Bó Fraích* and their modern cognates.

OIr. lemma	Old Irish form	stand. contemp. Modern Irish cognate
ad·ella	ad+PV1+ell+VROOT+W1 +PRT +PASS+CONJ+3P+SG aidleth	tadhaill+Verb+VTI +PastInd +Auto tadhlaíodh
	ad+PV1+ell+VROOT+W1 +IPF +CONJ+3P+SG aidleth	tadhaill+Verb+VTI +PastImp+Len thadhlaíodh
	ad+PV1+ell+VROOT+W1 +PAST+SUBJ +CONJ+3P+SG aidleth	tadhaill+Verb+VTI +PastSubj +3P+Sg tadhlaíodh
	ad+PV1+ell+VROOT+W1 +IMP +CONJ+3P+SG aidleth	tadhaill+Verb+VTI +Imper +3P+Sg tadhlaíodh
brissid	bris+VROOT+W2a +PRT +ABS+3P+SG brissis	bris+Verb+VTI +PastInd+Len bhris_sé/sí
	bris+VROOT+W2a +PRT +CONJ+2P+SG brissis	bris+Verb+VTI +PastInd+Len bhris_tú
do·léici	to+PV1+PROCL_JUNCT+lēc+VROOT+W2a +PRS+IND +CONJ+3P+SG dolléici	teilg+Verb+VTI +PresInd teilgeann_sé/sí
	to+PV1+PROCL_JUNCT+lēc+VROOT+W2a +PRS+IND +CONJ+2P+SG dolléici	teilg+Verb+VTI +PresInd teilgeann_tú
imm·ráidi	imbi+PV1+PRON+A+3P+SG+NEUT+PROCL_JUNCT+LEN+ro+AUG+rād+VROOT+W2a +PRT +CONJ+3P+SG immaroraid	do+Part+Vb iomráidh+Verb+VTI +PastInd+Len d'iomráidh_sé/sí
	imbi+PV1+PRON+A+3P+SG+NEUT+PROCL_JUNCT+LEN+ro+AUG+rād+VROOT+W2a +PRS+SUBJ +CONJ+1P+SG immaroraid	iomráidh+Verb+VTI PresSubj iomráidhe_mé
marbaid	marb+VROOT+W1 +IMP +CONJ+1P+PL marbam	maraigh+Verb+VTI +Imper +1P+Pl maraímis
	marb+VROOT+W1 +PRS+IND +CONJ+1P+PL marbam	maraigh+Verb+VTI +PresInd +1P+Pl maraímid
	marb+VROOT+W1 +PRS+SUBJ +CONJ+1P+PL marbam	maraigh+Verb+VTI PresSubj +1P+Pl maraímid

I have in a very preliminary fashion juxtaposed five verb forms from *Táin Bó Fraích* in Table 6.1. Only one of each interpretation is the specific inflected form found in *Táin Bó Fraích*. The Old Irish lexical-level analyses are also not exhaustive; only the most distinctive parses are given here. For example, an 'invisible' consonant mutation—although all outputted by the Old Irish FST—has not been considered, and relative readings are ignored (the form *dolléici* occurs in sentence-initial position (with an initial capital letter) in the text and cannot therefore be a (nasalising) relative form in the first place). The past subjunctive has been marked in yellow as it is not part of the current version of the Modern Irish FST. The past subjunctive has been absorbed in the contemporary language mainly by the conditional. For an overview of the subjunctive in Irish cf. McQuillan (2002).

Some problems can be observed. Old Irish only had synthetic verb endings, whereas (standardised) Modern Irish only partially has synthetic verb forms left (with a complete reworking of the ending sets). A 3sg. form in Old Irish often maps to a tense/mood ending underspecified for person/number in contemporary Modern Irish, as synthetic forms have been replaced largely by a syntagm of sg. verb plus pronoun in Modern Irish. Mapping between the abundant Early Irish 3sg. forms and modern forms is therefore not straightforward (except with modern synthetic verb forms, e.g., imp. 1pl. *maraímis* 'let us kill'). Further 'compatibility' issues include both a singular and plural passive in Old Irish, with only one autonomous form in Modern Irish.

Although a small set of modern verbs (the irregular verbs) echo the allomorphic divergence between independent and dependent formations in earlier Irish (e.g., (*do-*) *rinne*, *-dearna* 'did' < 3sg. *do·rigéni*, *·dergéni* 'has done'), features such as deuterotonic and prototonic, and absolute and conjunct endings—the latter representing a crucial distinction between independent and dependent[1] with simple verbs in Early Irish—have been gradually lost in Modern Irish. The ending distinction in prs. ind. 1pl. *marbmai* 'we kill' and its dependent counterpart *ní·marbam* 'we do not kill' (both modern *maraímid*), for example, has no significance in Modern Irish, although lenition (*ní m**h**araímid*), caused by *ní*, can be considered a feature of a dependent form in this case. The problem is that lenition does not automatically equal dependency: *bhris sé* 'he broke', for example, is not dependent. Moreover, in Old Irish, a conjunction such as *má* 'if' lenites a following independent form.

The *ro*-forms and unaugmented forms are conflated into one modern tense/mood, and the modern past tense (+PastInd) does not derive from the Old Irish unaugmented preterite (+PRT) but mostly from Old Irish augmented preterites. *Bhris sé/sí* '(s)he broke', for example, goes back, via pre-standard *do bhris*, to *ro·bris* (not to unaugmented preterite *brissis* in Table 6.1). Even in dependent constructions where *ro* 'survives', e.g, in modern *níor*, *ar*, there is absolutely no 'perfective' meaning; the particle accompanies a simple past tense (i.e., *níor bhris (sé/sí)* is closer in meaning to Old Irish unaugmented preterite *ní·bris* 'did not break' than to augmented preterite *ní·robris* 'has not broken', while the latter is its etymological predecessor). The 'bare' preterite is lost in (later) Modern Irish (cf. section 2.5). A workable and justified solution is to

[1] Apart from the imperative, which invariably has conjunct endings.

link the modern past indicative to both the 'bare' and augmented preterite.

Even the small subset of verb forms discussed above shows that one often deals with a one-to-many relationship from the viewpoint of Modern Irish. Instead of painfully trying to define the longest substring match between an Old Irish lexical-level tag and the equivalent Modern Irish one, one can work with a system whereby a *minimum* amount of lexical-level string elements should match. This translates into a step-wise isolation of correct upper-level string matches in the following way, using three examples from Table 6.1 in (28), (29) and (30) (Old Irish : Modern Irish):

(28) `{ad+PV1}` *and* `{ell+VROOT}` : `{tadhaill+Verb}`
`{+PRT}` : `{+PastInd}`
`{+PASS}` : `{+Auto}`
lower-level matches:
`{ad·ellath, ad·ellad, atom·ellad, ... aidleth, aidled, ním·aidled, ... }` :
`{tadhlaíodh}`

(29) `{imbi+PV1}` *and* `{rād+VROOT}` : `{iomráidh+Verb}`
`{+PRT}` : `{+PastInd}`
lower-level matches, including *ro*-forms (augmented):
`{imm·roraid, imma·roraid, imm·ráid, immus·ráid, imm·ráidi, ... }` :
`{d'iomráidh}`

(30) `{marb+VROOT}` : `{maraigh+Verb}`
`{+IMP}` : `{+Imper}`
`{+1P+PL}` : `{+1P+Pl}`
lower-level matches:
`{marbam}` : `{maraímis}`

Although the above exploratory 'mapping experiment' is an interesting linguistic exercise, and, if implemented, will undoubtedly assist work in, say, the development of synthetic to analytic verb formation in the history of Irish, a mapping using verb stem/root only, which is easier to implement (only involving mappings between Old Irish roots or preverb-plus-roots and modern stems), perhaps serves equally important purposes.

As Figure 6.1 shows, the linking process is bidirectional; one can also go from Modern Irish back to Old Irish. Verbs such as *lig* and *teilg* both derive from verbs with the Old Irish verb root *lēc*, but this connection became 'clouded' after the univerbation of the Early Irish compound *do·léici*, *·teilci* (*to-lēc-*) into *teilcid.* The unambiguous lexical-level string *lēc* in the Old Irish FST facilitates generation of the complete paradigm for verbs with root *lēc*. By means of diachronic lexical-level mappings, one is able to find out, via Old Irish, that modern *teilg*, for example, is related to *lig*, as illustrated in (31).

(31)
$$\begin{tikzcd}
\texttt{\{l\=ec+VROOT\}} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & \texttt{\{lig+Verb\}} \\
\texttt{\{to+PV1+l\=ec+VROOT\}} \arrow[r, leftrightarrow] & \texttt{\{teilg+Verb\}}
\end{tikzcd}$$

A similar approach can solve mappings that are not one-to-one, e.g., modern *tadhaill* 'touch', etc., does not go back to *ad·ella* (*ad-ell-*) but to *do·aidlea* (*to-ad-ell-*).[2] The Old Irish compounds do, however, share the same root *ell-*, a common denominator which can be used to track down all the compounds with this root element in Old Irish to find out what the origin of *tadhaill* might be, and which historical variants one might expect to encounter. 'Plain' lemmatisation to eDIL is perhaps more informative here, seeing that de Bhaldraithe (1981: 71) lists the unambiguous correspondence *tadhlaíonn* : *do-aidlea*, which in `droichead` is encoded as `tadhaill_v`, `tadhaill`, `br`, `17118`, `do-aidlea`.[3]

Being able to generate diachronic paradigms for cognate verb lemmas could be a very interesting and meaningful contribution to diachronic linguistics for Irish. This undoubtedly provides interesting insights into the development of verbs which originally had the same verb root, as well as providing new ways to systematically investigate lexicalisation of preverb(s) plus verb root (univerbation) in the historical development of the Irish language.

6.4.3 Lemmatisation and eDIL

Using Dereza (2016), an Early Irish inflected (verb) form can be linked its lemma. It is only a small step to connect this lemma, which originally came from eDIL, with the headword in the latter resource. The unique stable links for each headword can be employed to this end, as shown in Figure 6.1.

The tool `droichead`[4] (Scannell 2018) is a digitised version of the index prepared by de Bhaldraithe (1981), listing the Early Irish equivalent for entries in *FGB* (Ó Dónaill 1977). Kevin Scannell converted this index of mappings to a machine-readable list and augmented it with POS tags and the identification codes of the stable links to the eDIL headword. While de Bhaldraithe (1981) gives the prs. ind. 3sg. of the modern headword so that it matches the inflection given in DIL, Scannell employs the imp. 2sg., following *FGB*. The mapping *tadhaill* : *do·aidlea* has been discussed above. Another one is `teilg_v`, `teilg`, `br`, `18012`, `do-lēci`.

The FST for Modern Irish (Uí Dhonnchadha & van Genabith 2006) contains modern, contemporary lemmas based on *FGB* on the upper, lexical level. All inflected forms of a lemma could be extracted from this transducer and linked to their lemma in a table. As illustrated in Figure 6.1, combining this table with `droichead` provides us with a way to link Modern Irish inflected forms, via their (*FGB*) lemma, to eDIL.

[2] Cf. `http://dil.ie/17118`.

[3] `br` = *briathar*, 'verb'.

[4] Available at `https://github.com/kscanne/droichead`.

6.4.4 Normalisation

A normalisation method for Old Irish is proposed in Figure 6.1 using Dereza (2016). This method is somewhat analogous to the standardisation employed in the context of *Corpas Stairiúil na Gaeilge*, in which pre-standard Irish forms are mapped to their standardised equivalent using Scannell (2008) (cf. also Section A.1.2.4). The Early Irish Lemmatiser developed by Dereza (2016) contains a prediction component that can be considered a standardiser or normaliser: a form in a text—if not already in the dictionary of the Lemmatiser—is compared to inflected forms in the dictionary by approximate matching (a variant of the Levenshtein edit distance method, cf. section 3.3.2 and section 3.6.1). On the basis of this string similarity algorithm, the Lemmatiser then decides which known inflected form is closest to the unknown string—and subsequently provides the eDIL headword based on known form-lemma mappings in its dictionary. The Lemmatiser's dictionary consists of `{known string}` : `{lemma}` correspondences and can, on the basis of a wordlist/text, provide a list of triplets of the form `{unknown string}` : `{predicted known string}` : `{lemma}`.

The morphological analyser developed in the context of my project (cf. Chapter 4) generates full paradigms of verbs; it will therefore, in time, greatly surpass the amount of Old Irish inflected verb forms found in eDIL. Moreover, my FST contains (approximately) normalised forms. To both increase the power of the Lemmatiser and facilitate normalisation, surface-level forms generated by my FST can be added to the list of `{known string}` : `{lemma}` mappings in the Lemmatiser's dictionary. The likelihood of a form occurring in an Early Irish text that 'matches' a known form is now much greater, and the Lemmatiser prediction algorithm will result in much more accurate matches. The Lemmatiser's edit-distance algorithm is the basis for a normaliser in conjunction with the FST: the algorithm predicts a normalised form (lower-level/surface string originally from the FST) closest to the input form, and a full grammatical parse is returned thanks to the upper-level tags accompanying those predicted normalised strings in the FST.

6.4.5 FST adaptation and sequential transducers

In addition to normalisation (cf. section 6.4.4), spelling rules could be implemented in the lower-level rule framework of the morphological FST for Old Irish. An important question is to what extent—if at all—orthographical variation should be encoded as part of the FST (potentially starting from unambiguous symbols such underlying phonemes; cf. sections 4.3.4, 4.6.4.3 and 5.6.4). It is probably wise to have a distinct set of replace rules to create a 'canonical' version of the transducer containing only normalised forms, as opposed to a 'non-canonical' one with relaxation of spelling rules. If one wanted to generate inflectional paradigms without every possible spelling variant, for example, one can resort to a 'clean' FST with normalised (and perhaps somewhat idealised) spelling conventions.

Beesley & Karttunen (2003: Chapter 9) describe the sequential application of different transducers using the `lookup` tool (in our case, `flookup`) in combination with a script. Following their examples, there can be a separate transducer for capitalisation normalisation, one that relaxes accentuation rules (in languages that have diacritics), and one that handles general relaxation of spelling rules (including wrong spellings). In a script it can be specified, for example, that the 'canonical' transducer should apply first; if its fails to provide an analysis for an input word, the 'less canonical' transducer is tried, etc.

One can envisage an FST that relaxes the orthographical rules of normalised Old Irish. For example, orthographical variants of the type *fodlid* (*fodlaid*), *lécit* (*léicit*) and *rádid* (*ráidid*), discussed in section 5.6.4, can be produced by making the explicit marking of palatalised and non-palatalised consonants/consonant clusters optional, i.e., deleting *i* before a consonant (cluster) followed by a front vowel, or deleting *a* preceding *i* in a non-initial syllable (reflecting ə). It is during the development of one or more subsequent transducers that one may also think about variation in quality of the stem-final consonant with W2 verbs such as *ar·áili* and *fo·ruimi*, discussed in section 5.6.2. In such a 'non-canonical' or 'variant' lexical transducer, an upper-level tag such as `+VAR` for 'variant' can be prefixed or suffixed. Alternatively, the upper-level tag `+NORM` can be added to all forms in the normalised/canonical transducer.[5]

An open question is to what extent it is necessary to adapt the Old Irish FST to cover Middle Irish forms. Middle Irish grammatical features such as changes to verb stems and inflectional patterns go beyond orthography. In other words, normalisation by approximate matching (section 6.4.4), and relaxation of spelling rules by devising sequential transducers, are not expected to cover all variation encountered in Early Irish texts. FST adaptation might entail creating a new `lexc` architecture, either incorporated in the Old Irish FST or as part of a separate transducer. Such a new framework could contain, e.g., stems for univerbated compound verbs[6] and innovative ending sets. Instead of `+VAR`, the generated 'variants' could be encoded with an upper-level tag `+MID_IRISH`. These tags can be used to either exclude Middle Irish forms (e.g., if the purpose is to generate strictly Old Irish paradigms only) or include Middle Irish forms (e.g., to calculate the contribution of Middle Irish forms in a text).[7]

An important part of an FST not touched on in this thesis is a so-called morphological guesser (Beesley & Karttunen 2003: 444–451). Basically, this technique involves defining abstract and phonologically *possible* morphemes as part of the word in the lower level, typically a template for a word's stem, encoded on the upper level as such (e.g., `+GUESS_VSTEM`). As the FST 'knows' the inflectional rules, prefixes, etc., it guesses part of the word according to the abstract template defined and returns a string like `+GUESS_VSTEM+PRS+IND+1P+SG`. The linguist can extract and inspect all lower-level strings with the accompanying guessed tag and,

[5] Relaxing the rules will introduce inadvertent ambiguity with verbs of the type *dálaid* (W1) 'meets' vs. *dáilid* (W2a) 'bestows', in which allophonic variation with the stem-final consonant *l* is found (cf. `dil.ie/14357` and `dil.ie/14201`). A subsequent 'variant' transducer will generate identical 'underspecified' spellings such as prs. ind. 3sg. abs. *dálid* for both verbs. The resulting ambiguity cannot be catered for in the morphological analysis stage; one could consider semantic disambiguation strategies later in the pipeline, based on either manual checking or automatic techniques (e.g., verb-specific valency patterns, collocations).

[6] The process of various elements merging into a single lexical item, cf. section 2.5.

[7] Such a strict division between Old and Modern Irish tagged features is perhaps somewhat artificial; the univerbation of compound verbs, for example, can already be partly observed in Old Irish.

on the basis of this output, can identify stem entries not yet incorporated into the FST. When the concept of stem is not that clear-cut, as in the case of Old Irish verbs, the guessed output may also point to a logical stem entry (string) for a verb. In other words, a morphological guesser may be part of the linguistic analysis and discovery process relative to the language under investigation.

The FST and Part-Of-Speech Tagger for Modern Irish (Uí Dhonnchadha & van Genabith 2006) incorporates such a guesser. This means that a word will never end up not receiving a morphological parse. This explains why Figure 6.1 does not show the 'no result' on the Modern Irish side of things: if the word is a pre-standard variant, a morphological parse is assisted by the standardisation rules in *An Caighdeánaitheoir* (Scannell 2008); otherwise the guesser component as part of modern-language FST will retrieve an analysis. I have not yet implemented such a guesser, which means that there is a possibility that a verb form will not be recognised if it is not contained in the FST, and cannot be related to an inflected variant in Dereza's (2016) Lemmatiser dictionary.

6.5 Concluding remarks

Although the case study in Chapter 5 has greatly informed the challenges of working with forms that do not adhere to 'canonical' Old Irish, including Middle Irish ones, a logical next step is to consolidate the FST and focus on a narrower time frame. Closer collaboration with the *Chronologicon Hibernicum* project (section 1.5) and incorporation of contemporaneous Old Irish parsed data (illustrated in Figure 6.1) is part of the envisaged future activities.

Due to my focus on Old Irish, limitations of the present work include the fact that little attention has been paid to the 'middle period', i.e., anything between Old and contemporary Modern Irish. While it is my conviction that my methodological choices are justified, this thesis does not deal with insightful developments in Middle and Early Modern Irish and the intermediate verb forms that really connect up the dots. Moreover, establishing the correct linkages for a subset of verbs will undoubtedly be quite challenging, for various reasons. For example, some verbs are attested only in Modern or medieval Irish, but not in both. Others are formally identical (according to diachronic morphological rules) but etymologically unrelated (e.g., modern denominative *léasaigh* 'lease' (not in the index compiled by de Bhaldraithe (1981)) vs. *lésaigid*[8] 'illumines'. In other words, semantic as well as morphological examination (e.g., by means of a manual checking procedure) would be required for each verb, both in the lexicons and in the texts. This is a far from trivial problem.

The reader will easily appreciate that systematically linking all intermediate variants, in a one-man project such as the one described here, is impossible on the grand scale of the entire history of the language. It should be stressed, however, that the apparent disregard for intermediate forms between Old and Modern is ultimately and fundamentally inherent to my methodology. The very concept of a 'two-pronged attack' (Figure 3.3) is based on two 'anchor

[8] `http://dil.ie/30001`.

points', i.e. a morphosyntactic tagger for Old Irish and one for contemporary Modern Irish. Intermediate and variant forms are oriented towards the normative and well-resourced periods of Old and (contemporary) Modern Irish, by using standardisation/normalisation methods directed to either of the two taggers. This method was inspired by the *modus operandi* employed in the *Foclóir Stairiúil na Gaeilge* project (cf. Appendix A.1.1.6), which employs a standardiser rather than a newly created POS tagger specifically for pre-standard forms, with limited re-usability (Uí Dhonnchadha et al. 2014). In other words, it is expected that there is no need for building separate morphological (and, subsequently, morphosyntactic) parsers from the ground up for each and every historical variety of Irish (although one might consider adapting the 'normative' Old Irish FST, cf. section 6.4.5). It is my firm belief that this bidirectional approach will prove fundamental to bridging the Old and Modern Irish period: my morphological analysis tools, accompanied by normalisation approaches, will in due course be able to deal with Early Irish texts, and the 'modern' tools can be adapted to Early Modern Irish texts (c. 13th–mid 17th centuries).

Indeed, the tools for modern language are already successfully applied to a corpus of bardic poetry, bringing Classical Modern Irish into the field of Digital Humanities (Mac Cárthaigh 2018), cf. section A.1.2.3 in Appendix A. The highly standardised nature of texts for the latter period (or, better, genre) might provide an intermediate anchor point, both in linguistic and computational terms. This, however, can only be established by further developing the two-way adaptation method, which will hopefully slowly seal the 'lexicographical gap', an academic endeavour greatly welcomed by Irish historical linguists, philologists and computational linguists alike.

6.6 Synthesis

In addition to concluding the work, this chapter has introduced proposed ways of mapping between Old and Modern Irish verb forms, schematically represented in Figure 6.1. The core resources in both 'linking routes' are the morphological FSTs. Building a morphological FST for Old Irish has been the main objective of this thesis. Linking the FSTs reflects an interesting linguistic enterprise as morphological parses are retained in the mapping process, facilitating detailed mappings between Old and Modern Irish verb forms. Implementing lexical-level tag mappings between the Old and Modern Irish transducers facilitates the juxtaposition of historical paradigms and, subsequently, enables a study of the historical development of the Irish verb. It should be borne in mind, though, that a mapping, due to innovative processes and grammatical simplification, often involves a one-to-many relationship from the viewpoint of Modern Irish. A resource pivotal to lemmatisation is `droichead` (Scannell 2018), based on (de Bhaldraithe 1981). There are many linguistic challenges in creating an exhaustive mapping and lemmatisation framework, some of which have been discussed in section 6.5 in relation to formally identical but etymologically and semantically unrelated headwords.

Computational solutions to handling spelling and other types of linguistic variation, which plague virtually any NLP project dealing with non-standard language, have also been duly considered and reported on. The Lemmatiser developed by Dereza (2016) can assist in predicting normalised forms for Early Irish variants (section 6.4.4). Adapting the Old Irish FST framework (section 6.4.5) will subsequently enhance word recognition, and the application of sequential transducers allows for modularity and flexibility in a finite-state system, recommended as part of a project's linguistic planning phase (Beesley & Karttunen 2003: 283–293). While the present work has a strong focus on Old Irish, it has undoubtedly made advances in meaningfully defining and computationally encoding the verb stem in this normative language period. It is expected that this work will make a diachronic linking resource a less distant future reality, thereby accelerating scholarly endeavours in identifying historical Irish cognate verb forms.

Bibliography

Ahlqvist, Anders. 1994. Litriú na Gaeilge. In Kim McCone, Damian McManus, Cathal Ó Háinle, Nicholas Williams & Liam Breatnach (eds.), *Stair na Gaeilge: in ómós do Phádraig Ó Fiannachta*, 23–59. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

Antworth, Evan L. 1991. Introduction to two-level phonology. *Notes on Linguistics* 53. 4–18. https://software.sil.org/pc-kimmo/two-level-phonology/.

Baron, Alistair & Paul Rayson. 2008. VARD2: A tool for dealing with spelling variation in historical corpora. In *Postgraduate conference in corpus linguistics, 22 May 2008*. http://eprints.lancs.ac.uk/41666/.

Bauer, Bernhard. 2014. *The online database of the Old Irish Priscian glosses*. Indogermanistik Wien. https://www.univie.ac.at/indogermanistik/priscian/.

Beesley, Kenneth R. & Lauri Karttunen. 2003. *Finite-state morphology: Xerox tools and techniques*. Center for the Study of Language & Information.

Bergin, Osborn (ed.). 1930. *Sgéalaigheacht Chéitinn: Stories from Keating's history of Ireland*. 3rd edn. Dublin: Royal Irish Academy.

Bergin, Osborn. 1946. Irish grammatical tracts: III and IV. *Ériu* 14. Supplement, 167–257.

Bird, Steven, Ewan Klein & Edward Loper. 2009. *Natural Language Processing with Python: Analyzing text with the Natural Language Toolkit*. Sebastopol, CA: O'Reilly Media. http://www.nltk.org/book/.

Bollmann, Marcel. 2012. (semi-)automatic normalization of historical texts using distance measures and the Norma tool. In *Proceedings of the second workshop on annotation of corpora for research in the humanities (ACRH-2)*, 3–14. Lisbon, Portugal. https://www.linguistics.ruhr-uni-bochum.de/comphist/pub/acrh12.pdf.

Bollmann, Marcel, Florian Petran & Stefanie Dipper. 2014. Applying rule-based normalization to different types of historical texts—an evaluation. In Zygmunt Vetulani & Joseph Mariani (eds.), *Human Language Technology Challenges for Computer Science and Linguistics. LTC 2011* (Lecture Notes in Computer Science 8387), 166–177. Cham: Springer. https://doi.org/10.1007/978-3-319-08958-4_14.

Booij, Geert. 2012. *The grammar of words: an introduction to linguistic morphology*. 3rd edn. (Oxford Textbooks in Linguistics). Oxford: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199226245.001.0001.

Borin, Lars & Markus Forsberg. 2011. A diachronic computational lexical resource for 800 years of Swedish. In Caroline Sporleder, Antal van den Bosch & Kalliopi Zervanou (eds.), *Language technology for cultural heritage: Selected papers from the LaTeCH workshop series* (Theory and Applications of Natural Language Processing), 41–61. Berlin & Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-20227-8_3.

Breatnach, Liam. 1977. The suffixed pronouns in Early Irish. *Celtica* 12. 75–107.

Breatnach, Liam. 1994. An Mheán-Ghaeilge. In Kim McCone, Damian McManus, Cathal Ó Háinle, Nicholas Williams & Liam Breatnach (eds.), *Stair na Gaeilge: in ómós do Phádraig Ó Fiannachta*, 221–333. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

Brinton, Laurel J. & Elizabeth C. Traugott. 2005. *Lexicalization and language change*. 2nd edn. (Research Surveys in Linguistics). Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511615962.

Burdick, Anne, Johanna Drucker, Peter Lunenfeld, Todd Presner & Jeffrey Schnapp. 2012. *Digital Humanities*. London & Cambridge, MA: The MIT Press. https://www.dropbox.com/s/zcfhiphslciqe2k/9248.pdf?dl=1.

Busa, Roberto A. 2004. Foreword: Perspectives on the digital humanities. In Susan Schreibman, Ray Siemens & John Unsworth (eds.), *A companion to Digital Humanities*, xvi–xxi. Oxford: Blackwell Publishing. https://doi.org/10.1002/9780470999875.fmatter.

Carney, James (ed.). 1964. *The poems of Blathmac, son of Cú Brettan: together with the Irish Gospel of Thomas and a poem on the Virgin Mary* (Irish Texts Society 47). London: Irish Texts Society.

Corpas na Gaeilge 1600–1882. 2004. *The Irish Language Corpus. CD-ROM*. Dublin: Royal Irish Academy.

Crane, Gregory. 1991. Generating and parsing Classical Greek. *Literary and Linguistic Computing* 6(4). 243–245. https://doi.org/10.1093/llc/6.4.243.

de Bhaldraithe, Tomás. 1981. *Innéacs Nua-Ghaeilge don 'Dictionary of the Irish language'*. Baile Átha Cliath: Acadamh Ríoga na hÉireann.

Dereza, Oksana. 2016. Building a dictionary-based lemmatizer for Old Irish. In *Actes de la conférence conjointe JEP-TALN-RECITAL, vol. 6: CLTW*, 12–17. Paris. https://jep-taln2016.limsi.fr/actes/Actes%20JTR-2016/V06-CLTW.pdf.

Dinneen, Patrick S. (ed.). 1927. *Foclóir Gaeilge agus Béarla*. 2nd edn. New edition, revised and enlarged. Dublin: Irish Texts Society.

Doyle, Adrian, John P. McCrae & Clodagh Downey. 2019. A Character-Level LSTM Network Model for Tokenizing the Old Irish text of the Würzburg Glosses on the Pauline Epistles. In *Proceedings of the Celtic Language Technology Workshop 2019, Dublin, 19 Aug., 2019*, 70–79. https://docs.wixstatic.com/ugd/705d57_59525d529bef46fa9afe50df336832e2.pdf.

Drucker, Johanna. 2013. *Intro to Digital Humanities*. UCLA Centre for Digital Humanities. http://dh101.humanities.ucla.edu/.

Etxeberria, Izaskun, Iñaki Alegria, Larraitz Uria & Mans Hulden. 2016. Evaluating the Noisy Channel Model for the normalization of historical texts: Basque, Spanish and Slovene. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*, 1064–1069. Portorož, Slovenia: European Language Resources Association (ELRA). `http://www.lrec-conf.org/proceedings/lrec2016/pdf/147_Paper.pdf`.

FitzPatrick, Elizabeth. 2004. *Royal inauguration in Gaelic Ireland c. 1100–1600: A cultural landscape study* (Studies in Celtic history). Woodbridge, Suffolk: Boydell Press.

Fomin, Maxim & Gregory Toner. 2005. Digitizing a dictionary of medieval Irish: the eDIL project. *Literary and linguistic computing* 21(1). 83–90. `https://doi.org/10.1093/llc/fqh050`.

Forsberg, Markus. 2007. *Three tools for language processing: BNF converter, Functional Morphology, and Extract*. Ph.D. thesis. Chalmers University of Technology & Göteborg University. `https://svn.spraakdata.gu.se/repos/markus/pub/phd2007_print_version.pdf`.

Forsberg, Markus & Aarne Ranta. 2004. Functional morphology. In *Proceedings of the ninth ACM SIGPLAN international conference on functional programming (ICFP '04)*, 213–223. Snow Bird, UT: ACM. `https://doi.org/10.1145/1016850.1016879`.

Fransen, Theodorus. Forthcoming. Automatic morphological analysis and interlinking of historical Irish cognate verb forms. In Elliott Lash, Fangzhe Qiu & David Stifter (eds.), *Corpus-based approaches to morphosyntactic variation and change in medieval Celtic languages*. Berlin: De Gruyter.

Green, Antony. 1995. *Old Irish verbs and vocabulary*. Somerville, MA: Cascadilla Press.

Greene, David (ed.). 1955. *Fingal Rónáin and other stories*. Dublin: Dublin Institute for Advanced Studies.

Greene, David. 1958. The analytic forms of the verb in Irish. *Ériu* 18. 108–112.

Greene, David. 1966. *The Irish language* (Irish life and culture series). Dublin: Published for the Cultural Relations Committee of Ireland at the Three Candles, Limited.

Greene, David. 1973. Synthetic and analytic: A reconsideration. *Ériu* 24. 121–133.

Griffith, Aaron & David Stifter. 2007-2013. *A Dictionary of the Old Irish Glosses in the Milan Codex Ambrosianus C 301 inf.* Institut für Sprachwissenschaft, Universität Wien. `https://www.univie.ac.at/indogermanistik/milan_glosses/`.

Hockey, Susan. 2004. The history of Humanities Computing. In Susan Schreibman, Ray Siemens & John Unsworth (eds.), *A companion to Digital Humanities* (Blackwell companions to literature and culture 26), 1–19. Oxford: Blackwell Publishing. `https://doi.org/10.1002/9780470999875.ch1`.

Hopcroft, John. E., Rajeev Motwani & Jeffrey D. Ullman. 2001. *Introduction to Automata Theory, Languages and Computation*. 2nd edn. Addison-Wesley. https://mcdtu.files.wordpress.com/2017/03/introduction-to-automata-theory.pdf.

Huang, Lian, Yinan Peng, Huan Wang & Zhenyu Wu. 2002. Statistical Part-of-Speech Tagging for Classical Chinese. In Petr Sojka, Ivan Kopeček & Karel Pala (eds.), *Text, speech and dialogue*. 5th International Conference, TSD 2002, Brno, Czech Republic September 9–12, 2002. Proceedings (Lecture Notes in Computer Science; Vol. 2448: Lecture notes in artificial intelligence), 115–122. Berlin: Springer. https://doi.org/10.1007/3-540-46154-X_15.

Huet, Gérard. 2003. Towards computational processing of Sanskrit. In S. M. Rajeev Sangal Bendre & Udaya Narayana Singh (eds.), *International Conference on Natural Language Processing (ICON)*, 40–48. Mysore, India. http://gallium.inria.fr/~huet/PUBLIC/icon.pdf.

Huet, Gérard. 2005. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *Journal of Functional Programming* 15(4). 573–614. https://doi.org/10.1017/S0956796804005416.

Hughes, Art. 2008. *Leabhar mór bhriathra na Gaeilge: The Great Irish Verb Book*. Béal Feirste: Clólann Bheann Mhadagáin.

Hulden, Mans. 2009. Foma: a finite-state compiler and library. In *Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics*, 29–32. https://dl.acm.org/citation.cfm?id=1609057.

Indurkhya, Nitin & Fred J. Damerau (eds.). 2010. *Handbook of Natural Language Processing*. 2nd edn. (Chapman & Hall/CRC machine learning & pattern recognition series). Boca Raton, London & New York: CRC Press.

Johnson, Charles Douglas. 1972. *Formal aspects of phonological description*. The Hague: Mouton. http://idiom.ucsd.edu/~bakovic/compphon/Johnson%201972%201-up.pdf.

Jurafsky, D. & J.H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. 2nd edn. (Prentice Hall series in artificial intelligence). Upper Saddle River, New Jersey: Pearson Prentice Hall.

Jurish, Bryan. 2008. Finding canonical forms for historical German text. In Angelika Storrer, Alexander Geyken, Alexander Siebert & Kay-Michael Würzner (eds.), *Text resources and lexical knowledge: Selected papers from the 9th conference on Natural Language Processing KONVENS 2008* (Text, Translation, Computational Processing [TCCP] 8), 27–37. Berlin & New York: Mouton de Gruyter.

Jurish, Bryan. 2010. Comparing canonicalizations of historical German text. In Jeffrey Heinz, Lynne Cahill & Richard Wicentowski (eds.), *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, 72–77. http://www.aclweb.org/anthology/W10-2209.

Kavanagh, Séamus. 2001. *A lexicon of the Old Irish glosses in the Würzburg manuscript of the epistles of St. Paul*. Dagmar S. Wodtko (ed.) (Mitteilungen der Prähistorischen Kommission 45. Lexika und Fachwörterbücher). Wien: Österreichische Akademie der Wissenschaften.

Kay, Martin. 2003. Introduction. In Ruslan Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, xvii–xx. Oxford: Oxford University Press.

Kelleher, John. 2016. *Fundamentals of Machine Learning for Neural Machine Translation. Presented at Translating Europe Forum 2016: Focusing on Translation Technologies*. European Commission Directorate-General for Translation. `https://doi.org/10.21427/D78012`.

Koehn, Philipp. 2010. *Statistical Machine Translation*. Cambridge: Cambridge University Press. `https://doi.org/10.1017/CBO9780511815829`.

Koskenniemi, Kimmo. 1983. *Two-level morphology: A general computational model for word-form recognition and production.* (Publications of the Department of General Linguistics, University of Helsinki 11). Helsinki: University of Helsinki. `http://www.ling.helsinki.fi/~koskenni/doc/Two-LevelMorphology.pdf`.

Kyle P. Johnson, et al. 2014–2017. *CLTK: The Classical Language Toolkit*. `https://github.com/cltk/cltk`.

Lash, Elliott. 2014a. Subject positions in Old and Middle Irish. *Lingua* 148. 278–308.

Lash, Elliott. 2014b. *The Parsed Old and Middle Irish Corpus (POMIC)*. Version 0.1. `https://www.dias.ie/celt/celt-publications-2/celt-the-parsed-old-and-middle-irish-corpus-pomic/`.

Le Mair, Esther. 2011. *Secondary verbs in Old Irish: A comparative-historical study of patterns of verbal derivation in the Old Irish glosses*. Department of Old and Middle Irish and Celtic Philology, National University of Ireland, Galway. `http://hdl.handle.net/10379/3113`.

Lynn, Teresa. 2012. Medieval Irish and Computational Linguistics. *Australian Celtic Journal* 10. 13–27. `https://www.computing.dcu.ie/~tlynn/Lynn-MedievalIrish.pdf`.

Mac Cárthaigh, Eoin. 2018. Research case studies: Bringing bardic poetry into the light. In *Trinity College Dublin Provost & President's Annual Review 2017/18*, 28–31. `http://www.tcd.ie/provost/review/2018/annualreview.pdf`.

Mac Coisdealbha, Pádraig. 1998. *The syntax of the sentence in in Old Irish: Selected studies from a descriptive, historical and comparative point of view*. Graham Isaac (ed.) (Buchreiche der Zeitschrift für Celtische Philologie 16). Tübingen: Niemeyer.

Mahon, William J. 2006. Irish literature [5] 19th century. In John T. Koch (ed.), *Celtic culture: a historical encyclopedia*, vol. III, 1011–1014. Santa Barbara, CA: ABC-Clio.

McCarty, Willard. 2005. *Humanities Computing*. Basingstoke: Palgrave Macmillan.

McCone, Kim. 1985. The Würzburg and Milan Glosses: Our earliest sources of 'Middle Irish'. *Ériu* 36. 85–106.

McCone, Kim. 1994. An tSean-Ghaeilge agus a réamhstair. In Kim McCone, Damian McManus, Cathal Ó Háinle, Nicholas Williams & Liam Breatnach (eds.), *Stair na Gaeilge: in* *ómós do Phádraig Ó Fiannachta*, 61–219. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

McCone, Kim. 1997. *The Early Irish verb*. 2nd edn. (Maynooth monographs 1). Revised edition with *index verborum*. Maynooth: An Sagart.

McCone, Kim. 2005. *A first Old Irish grammar and reader: Including an introduction to middle irish* (Maynooth Medieval Irish Texts 3). Maynooth.

McCone, Kim, Damian McManus, Nicholas Ó Háinle Cathal Williams & Liam Breatnach (eds.). 1994. *Stair na Gaeilge: in ómós do Phádraig Ó Fiannachta*. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

McManus, Damian. 1994. An Nua-Ghaeilge Chlasaiceach. In Kim McCone, Damian McManus, Cathal Ó Háinle, Nicholas Williams & Liam Breatnach (eds.), *Stair na Gaeilge: in ómós do Phádraig Ó Fiannachta*, 335–445. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

McManus, Damian & Eoghan Ó Raghallaigh (eds.). 2010. *A Bardic miscellany: Five hundred Bardic poems from manuscripts in Irish and British libraries* (Trinity Irish Studies, 1). Dublin: Department of Irish, Trinity College, Dublin.

McQuillan, Peter. 2002. *Modality and grammar: A history of the Irish subjunctive* (Maynooth Studies in Celtic Linguistics 5). Maynooth: Department of Old Irish, National University of Ireland.

Meid, Wolfgang (ed.). 1974. *Táin Bó Fraích*. 2nd edn. (Mediaeval and Modern Irish Series 22). Dublin: Dublin Institute for Advanced Studies.

Meid, Wolfgang, Albert Bock, Benjamin Bruch & Aaron Griffith (eds.). 2015. *The romance of Froech and Findabair, or, The driving of Froech's cattle: Táin Bó Froích* (Innsbrucker Beiträge zur Kulturwissenschaft, Neue Folge 10). Innsbruck: Institut für Sprachen und Literaturen der Universität Innsbruck.

Mitkov, Ruslan (ed.). 2005. *The Oxford handbook of computational linguistics*. Oxford: Oxford University Press. `https://doi.org/10.1093/oxfordhb/9780199276349.001.0001`.

Mohri, Mehryar, Afshin Rostamizadeh & Ameet Talwalkar. 2012. *Foundations of Machine Learning* (Adaptive computation and machine learning series). London & Cambridge, MA: MIT Press.

Moon, Taesun & Jason Baldridge. 2007. Part-of-Speech Tagging for Middle English through alignment and projection of parallel diachronic texts. In *Proceedings of the 2007 joint conference on empirical methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 390–399. `http://www.aclweb.org/anthology/D07-1041`.

Ní Chatháin, Próinséas. 1969–1970. Léirmheas – Táin Bó Fraích. *Éigse* 13. 74–77.

Nyhan, Julianne. 2006a. *The application of XML to the Historical Lexicography of Old, Middle and Early Modern Irish: a Lexicon-based Analysis*. Ph.D. thesis. University College Cork. `http://epu.ucc.ie/theses/jnyhan/Nyhanthesisultima.html`.

Nyhan, Julianne. 2006b. The Digital Dinneen project: Further avenues for CELT. In *Conference abstracts of Digital Humanities 2006, the first ADHO international conference, 5-9 July 2006*, 153–154. Université Paris-Sorbonne: Centre de Recherche Cultures Anglophones et Technologies de l'Information. `http://allc-ach2006.colloques.paris-sorbonne.fr/DHs.pdf`.

Nyhan, Julianne. 2008. Developing integrated editions of minority language dictionaries: the Irish example. *Literary and Linguistic Computing* 23(1). 3–12. `https://doi.org/10.1093/llc/fqm038`.

Ó Cróinín, Dáibhí. 2001. The earliest Old Irish glosses. In Rolf Bergmann, Elvira Glaser & Claudine Moulin-Fankhänel (eds.), *Mittelalterliche volksprachige glossen: Internationale Fachkonferenz des Zentrums für Mittelalterstudien der Otto-Friedrich-Universität Bamberg 2. bis 4. August 1999* (Germanistische Bibliothek, 13), 7–31. Heidelberg: Winter.

Ó Crualaoich, Conchubhar. 1999. *Some irregular Syncope Patterns in Old Irish*. Unpublished Ph.D. thesis. National University of Ireland, Maynooth.

Ó Dónaill, Niall (ed.). 1977. *Foclóir Gaeilge-Béarla*. Irish-English Dictionary. Baile Átha Cliath: An Gúm.

Ó Donnaíle, Caoimhín. 2014. Tools facilitating better use of online dictionaries: Technical aspects of Multidict, Wordlink and Clilstore. In *Proceedings of the First Celtic Language Technology Workshop*, 18–27. Dublin, Ireland. `http://www.aclweb.org/anthology/W14-4603`.

Ó Háinle, Cathal. 2006. Irish literature [4] post-classical. In John T. Koch (ed.), *Celtic culture: A historical encyclopedia*, vol. III, 1005–1011. Santa Barbara, CA: ABC-Clio.

Ó hUiginn, Ruairí. 2013. Transmitting the text: some linguistic issues in the work of the Franciscans. In Raymond Gillespie & Ruairí Ó hUiginn (eds.), *Irish Europe, 1600–1650: Writing and learning*, 85–104. Dublin: Four Courts Press.

Ó Muircheartaigh, Peadar. 2015. *Gaelic dialects past and present: a study of modern and medieval dialect relationships in the Gaelic languages*. Ph.D. thesis. University of Edinburgh. `http://hdl.handle.net/1842/20473`.

O'Rourke, Alan J., Alexander M. Robertson, Peter Willett, Penny Eley & Penny Simons. 1997. Word variant identification in Old French. *Information Research* 2(4). `http://www.informationr.net/ir/2-4/paper22.html`.

Packard, David W. 1973. Computer-assisted morphological analysis of Ancient Greek. In *Proceedings of the 5th conference on computational linguistics*, vol. 2 (COLING '73), 343–355. Pisa, Italy: Association for Computational Linguistics. `https://doi.org/10.3115/992567.992595`.

Parodi, Teresa & Michael J. McCarthy. 2010. Morphology. In Kirsten Malmkjær (ed.), *The Routledge Linguistics Encyclopedia*, 3rd edn., 366–376. London & New York: Routledge.

Passarotti, Marco Carlo. 2010. Leaving behind the less-resourced status. The case of Latin through the experience of the Index Thomisticus Treebank. In Kepa Sarasola, Francis M. Tyers & Mikel L. Forcada (eds.), *7th SaLTMiL workshop on creation and use of basic* *lexical resources for less-resourced languages*, 27–32. http://www.lrec-conf.org/proceedings/lrec2010/workshops/W21.pdf.

Pedersen, Holger. 1909–13. *Vergleichende Grammatik der keltischen Sprachen*. 2 vols. Göttingen: Vandenhoeck & Ruprecht.

Piotrowski, Michael. 2012. *Natural language processing for historical texts* (Synthesis Lectures on Human Language Technologies. Vol. 5, No. 2). Morgan & Claypool Publishers. https://doi.org/10.2200/S00436ED1V01Y201207HLT017.

Quin, E. G. (ed.). 1983. *Dictionary of the Irish language: Based mainly on Old and Middle Irish materials*. Compact edition. Dublin: Royal Irish Academy.

Rayson, Paul, Dawn E. Archer, Alistair Baron, Jonathan Culpeper & Nicholas Smith. 2007. Tagging the Bard: Evaluating the accuracy of a modern POS tagger on Early Modern English corpora. In Matthew Davies, Paul Rayson, Susan Hunston & Pernilla Danielsson (eds.), *Proceedings of the Corpus Linguistics conference (CL2007), 27-30 July 2007*. UK. https://e-space.mmu.ac.uk/619647/.

Robertson, Alexander M. & Peter Willett. 1992. Searching for historical word-forms in a database of 17th-century English text using spelling-correction methods. In Nicholas Belkin, Peter Ingwersen & Annelise Mark Pejtersen (eds.), *Proceedings of the 15th annual international ACM SIGIR conference on research and development in Information Retrieval*, 256–265. New York, NY: ACM. https://doi.org/10.1145/133160.133208.

Rögnvaldsson, Eiríkur & Sigrún Helgadóttir. 2011. Morphosyntactic tagging of Old Icelandic texts and its use in studying syntactic variation and change. In Caroline Sporleder, Antal van den Bosch & Kalliopi Zervanou (eds.), *Language technology for cultural heritage: Selected papers from the LaTeCH workshop series* (Theory and Applications of Natural Language Processing), 63–76. Berlin & Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-20227-8_4.

Rossiter, Trudy. 2004. *Verbal Composition in Old Irish with Special Reference to Multi-preverb Compounds*. Unpublished Ph.D. thesis. National University of Ireland, Maynooth.

Russell, Paul. 1995. *An introduction to the Celtic languages*. London & New York: Longman.

Russell, Paul. 2005. What was best of every language: The early history of the Irish language. In Dáibhí Ó Cróinín (ed.), *A new history of Ireland*. Vol. 1: *Prehistoric and early Ireland*, 405–450. Oxford: Oxford University Press.

Russell, Paul. 2006. The Irish language. In John T. Koch (ed.), *Celtic culture: A historical encyclopedia*, vol. III, 985–993. Santa Barbara, CA: ABC-Clio.

Scannell, Kevin. 2008. *An Caighdeánaitheoir*. https://github.com/kscanne/caighdean/.

Scannell, Kevin. 2017. *Caighdeánú na Gaeilge. Paper presented at the 2017 ACIS national meeting, 01-04-2017*. Kansas City, Missouri. http://cs.slu.edu/~scannell/pub/acis17-paipear.pdf.

Scannell, Kevin. 2018. *droichead: Nascanna idir Foclóir Uí Dhónaill agus DIL*. https://github.com/kscanne/droichead.

Scheible, Silke, Richard J. Whitt, Martin Durrell & Paul Bennett. 2011. Evaluating an 'off-the-shelf' POS-tagger on Early Modern German text. In *Proceedings of the 5th ACL-HLT workshop on language technology for cultural heritage, social sciences, and humanities, LaTeCH '11, Portland, Oregon*, 19–23. Stroudsburg, PA: Association for Computational Linguistics. `http://dl.acm.org/citation.cfm?id=2107636.2107639`.

Schreibman, Susan, Ray Siemens & John Unsworth (eds.). 2004. *A companion to Digital Humanities* (Blackwell companions to literature and culture 26). Oxford: Blackwell Publishing. `https://doi.org/10.1002/9780470999875`.

Schumacher, Stefan. 2004. *Die keltischen Primärverben: Ein vergleichendes, etymologisches und morphologisches Lexicon*. Unter Mitarbeit von Britta Schulze-Thulin und Caroline aan de Wiel. Innsbruck: Institut für Sprachen und Literaturen.

Smith, Neel. 2016. Morphological analysis of historical languages. *Bulletin of the Institute of Classical Languages* 59(2). 89–102. `https://doi.org/10.1111/j.2041-5370.2016.12040.x`.

Souvay, Gilles & Jean-Marie Pierrel. 2009. LGeRM: Lemmatisation des mots en Moyen Français. *Traitement Automatique des Langues* 50(2). 149–172. `https://hal.archives-ouvertes.fr/halshs-00396452/`.

Stein, Achim. 2007. Resources and tools for Old French text corpora. In Yuji Kawaguchi, Toshihiro Takagaki, Nobuo Tomimori & Yoichiro Tsuruga (eds.), *Corpus-Based Perspectives in Linguistics* (Usage Based Linguistic Informatics 6), 217–229. Amsterdam & Philadelphia: John Benjamins. `https://doi.org/10.1075/ubli.6.15ste`.

Stifter, David. 2006. *Sengoídelc: Old Irish for beginners*. Syracuse, New York: Syracuse University Press.

Stifter, David. 2009. Early Irish. In Martin J. Ball & Nicole Müller (eds.), *The Celtic languages*, 2nd edn. (Routledge language family series), 55–116. Abingdon & New York: Routledge.

Stokes, Whitley & John Strachan (eds.). 1901–1910. *Thesaurus Palaeohibernicus: A collection of Old-Irish glosses, scholia, prose, and verse*. 3 vols. Cambridge: Cambridge University Press.

Strachan, John. 1949. *Old-Irish paradigms, and selections from the Old-Irish glosses*. 4th edn. Originally published in 1904/1905. Revised by Osborn Bergin, with notes and vocabulary. Dublin: Royal Irish Academy.

Thurneysen, Rudolf. 1946. *A grammar of Old Irish*. Trans. by Daniel A. Binchy & Osborn Bergin. Revised and enlarged edition. Dublin: Dublin Institute for Advanced Studies. Repr. 1993, with supplement.

Ua Súilleabháin, Seán. 2006. Dictionaries and grammars [1] Irish. In John T. Koch (ed.), *Celtic culture: A historical encyclopedia*, vol. II, 587–589. Santa Barbara, CA: ABC-Clio.

Uí Dhonnchadha, Elaine & Josef van Genabith. 2006. A Part-Of-Speech Tagger for Irish using Finite-State Morphology and Constraint Grammar Disambiguation. In *Proceedings of the 5th international conference on language resources and evaluation (LREC 2006)*, 2241–

2244. Genoa, Italy: European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2006/pdf/193_pdf.pdf.

Uí Dhonnchadha, Elaine, Kevin Scannell, Ruairí Ó hUiginn, Eilís Ní Mhearraí, Máire Nic Mhaoláin, Brian Ó Raghallaigh, Gregory Toner, Séamus Mac Mathúna, Déirdre D'Auria, Eithne Ní Ghallchobhair & Niall O'Leary. 2014. *Corpas na Gaeilge* (1882-1926): Integrating historical and Modern Irish text. In Kristín Bjarnadóttir, Mathew Driscoll, Steven Krauwer, Stelios Piperidis, Cristina Vertan & Martin Wynne (eds.), *Proceedings of the LREC 2014 workshop LRT4HDA: Language resources and technologies for processing and linking historical documents and archives - deploying linked open data in cultural heritage*, 12–18. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2014/workshops/LREC2014Workshop-LRT4HDA%20Proceedings.pdf.

Wilcock, Graham. 2009. *Introduction to linguistic annotation and text analytics* (Synthesis Lectures on Human Language Technologies. Vol. 2, No. 1). Morgan & Claypool Publishers. https://doi.org/10.2200/S00194ED1V01Y200905HLT003.

Williams, Nicholas. 1994. Na canúintí a theacht chun solais. In Kim McCone, Damian McManus, Cathal Ó Háinle, Nicholas Williams & Liam Breatnach (eds.), *Stair na Gaeilge: in ómós do Phádraig Ó Fiannachta*, 447–478. Maigh Nuad: Roinn na Sean-Ghaeilge, Coláiste Phádraig.

Wodtko, Dagmar S. 2007. Das Verb im Lexikon: Indogermanisch und Irisch. *International Journal of Diachronic Linguistics and Linguistic Reconstruction* 4(2). 91–133.

Appendix A

Survey of digital linguistic resources for historical Irish

A.1 Available

A.1.1 Lexicons

A.1.1.1 electronic Dictionary of the Irish language

Background and purpose — the electronic Dictionary of the Irish language (eDIL) is a digitised version of Dictionary of the Irish Language (Quin 1983). Work on eDIL started in 2003.[1] Opening up the wealth of information stored in the dictionary and making it accessible to a variety of users has been the central aim (Fomin & Toner 2005: 84). The initial objective of the work was not to revise the dictionary but to make it searchable online.

The original Dictionary of the Irish Language, on which eDIL is based, is sourced mainly from Old and Middle Irish sources (7th–12th centuries A.D.). At the time of compilation of the dictionary, however, Middle Irish was understood by many to extend beyond 1200 A.D. (Breatnach 1994: 221). Many inconsistencies in the paper edition have been incorporated in eDIL. Furthermore, the dictionary is not exhaustive in terms of inflectional forms provided. However, work on a revised electronic edition began in 2007, mainly based on publications in academic journals for the period 1932 to the present. The result of this accumulated into a revised, or second, electronic edition which was completed in 2013. According to the website, 'work has continued since then on primary sources and is expected to be completed by 2019'.[2]

Contents and mark-up — the digitised text has been marked up in eXtensible Markup Language (XML) following the guidelines of the Text Encoding Initiative[3] (TEI) for Print Dictionaries. The following discrete data types were identified and tagged accordingly:

[1] `http://www.dil.ie/`.

[2] `http://www.dil.ie/about`.

[3] `http://www.tei-c.org/`.

- headwords
- definitions
- internal cross-references
- grammatical information (case, stem, number, etc.)
- citations from medieval sources
- translations of citations
- source references, including title of work and page reference
- language of the text
- lemmas

Structural mark-up has been carried out automatically, whereas all linguistic mark-up was manual.[4] Parts-of-speech have been added where these have been determinable. While grammatical forms have been annotated (tense, person, number), inflectional forms are not contained within the grammatical tags, and invariably receive the tag `Ovar` (orthographical variant) in the XML (cf. Code Example A.1 in section A.2.1.1 below).

Accessibility and level of search available — a search option has been implemented, consisting of a basic and an advanced search. The latter can be used to restrict the search to some of the discrete data types listed above (e.g., headword, language) or specify grammatical information (Part-Of-Speech category, stem, tense, etc.).

A.1.1.2 Glosses databases

The Old Irish Glosses databases discussed below, as well as additional material, are currently being streamlined and prepared for online publication in the context of the *Chronologicon Hibernicum* project.[5]

A.1.1.2.1 Milan Glosses database

Background and purpose — a Dictionary of the Old-Irish Milan glosses (Griffith & Stifter 2007-2013)[6] was part of a series of related works dealing with the Old Irish glosses to Latin texts surviving in manuscripts on the European continent: the so-called Würzburg, Milan and Sankt-Gallen (Priscian) glosses, constituting our main body of surviving contemporary sources for Old Irish (8th–9th centuries A.D.) (Thurneysen 1946: 4-6).

[4] Prof. Gregory Toner, pers. comm. 15/02/2012.

[5] Introducing the *Chronologicon Hibernicum*. Paper presented at the 10th Celtic Linguistics Conference (CLC10), 4-5 September 2018,https://www.maynoothuniversity.ie/sites/default/files/assets/document/Celtic%20Linguistics%20Conference%20-%20Abstract%20Booklet_2.pdf.

[6] http://www.univie.ac.at/indogermanistik/milan_glosses.htm.

The Milan glosses constitute the largest corpus of the Old Irish glosses (8th and 9th century A.D.) (Thurneysen 1946: 5). The aim of the Milan glosses project was to facilitate research into the phonology, morphology, morpho-phonology and syntax of the language of this text, to provide 'a clearer picture of the state of the language at the beginning of the 9th century ... with consequences for future grammars, books and articles about Old Irish'.

Contents and mark-up — the corpus was implemented as a lexical tool with the relational database software `Filemaker`,[7] created out of various tables, including glosses with translations, a dictionary and sentence structure. A substantial part of the digitisation of the Milan glosses included the copy-typing and partial revision of the material published in *Thesaurus Palaeohibernicus* (Stokes & Strachan 1901–1910).[8]

Accessibility and level of search available — the database can be downloaded (PC or Mac) through the website. Various lay-outs can be chosen, the main one being the Database lay-out. A search option is available.

A.1.1.2.2 Priscian Glosses database

Background and purpose — The Online Database of the Old Irish Priscian Glosses[9] (Bauer 2014) is a corpus dictionary of all the Old Irish glosses dealing with the Latin grammar of Priscian. The project was part of a series of related works dealing with the Old Irish glosses to Latin texts surviving in manuscripts on the European continent.

Contents and mark-up — The corpus was implemented as a lexical tool with the relational database software `Filemaker` (cf. footnote 7), created out of various tables including glosses with translations, a dictionary and sentence structure. For the main corpus of the Priscian glosses, a pre-existing online database containing the full text of the St Gall glosses[10] was used.

Accessibility and level of search available — the database can be downloaded (PC or Mac) through the website. Various lay-outs can be chosen, the main one being the Database lay-out. A search option is available.

A.1.1.3 *In Dúil Bélrai*

Background and purpose — *In Dúil Bélrai* (Old Irish for 'The Glossary') provides an online English-Old Irish glossary and a database of 5,000 Old Irish conjugated verb forms.[11] Apart from offering a reverse search facility (English-Old Irish), the website assists Old Irish readers in finding the headword of an Old Irish inflected variant.

Contents and mark-up — the main (headwords) part of *In Dúil Bélrai* was manually extracted from the Dictionary of the Irish Language (DIL, cf. section A.1.1.1) by Dennis King,

[7] `http://www.filemaker.com/`.

[8] Pdf versions of the copy-typed text are available at `http://www.univie.ac.at/indogermanistik/milan_glosses.htm`.

[9] `http://www.univie.ac.at/indogermanistik/priscian/`.

[10] Published at `http://www.stgallpriscian.ie/`.

[11] Available at `http://www.smo.uhi.ac.uk/sengoidelc/duil-belrai/`.

who also included English definitions. The inflected verb forms together with their grammatical description were supplied by other members of the team,[12] from various sources. If the Old Irish word in question was found under a different headword in DIL, King noted that headword and it is recorded in the database. The verbforms table in *In Dúil Bélrai* provides a very partial lemmatisation table for Old Irish.[13]

The verb forms were implemented as lemmatisation tables in Wordlink, which links webpages word-by-word to online dictionaries, and in Multidict, an electronic application developed by Caoimhín P. Ó Donnaíle with a multiple dictionary lookup facility (Ó Donnaíle 2014) to interconnect online dictionaries.[14] Multidict incorporates a functionality to link to the electronic Dictionary of the Irish Language (eDIL).

The headword suggestion mechanism in Multidict, which can be prioritised in different ways, consists of the following elements (Ó Donnaíle 2014):

- spellchecking and affixation rules;
- lemmatisation tables. Ó Donnaíle (2014) reports that 1.4 million word forms reside in the `lemmas` table in the Multidict database;
- algorithmic lemmatisation: for example, removal of initial mutations.

Accessibility and level of search available — *In Dúil Bélrai* provides a search interface for English-Old Irish and accepts inflected verb forms, as well as headwords.

A.1.1.4 *Bunadas*

Background and purpose — *Bunadas* 'origin' (standard contemporary Modern Irish *bunús*) is an open web-based tool[15] for finding cognate word forms in the Celtic languages (as well as in Indo-European languages, Proto-Celtic and Proto-Indo-European).

Contents and mark-up — etymologically related word forms are encoded in clusters together with their language code. The program itself can 'travel' from word to group, and can find out connections with other clusters and establish long-term connections. There are 36,000 words in Bunadas.[16]

Accessibility and level of search available — a search box is available, accepting a word pattern, that allows searching bidirectionally for a large amount of Indo-European languages as well as Proto-Indo-European. A scale feature can be used to adjust the distance between the clusters, to allow for long-term connections, or to restrict the amount of cognates produced. Links to Multidict (cf. section A.1.1.3) have been added.[17]

[12] Liz Gabay and Elliott Lash. Liz Gabay also proofread the glossary work carried out by Dennis King (Liz Gabay, pers. comm. 03/07/2015).

[13] Caoimhín P. Ó Donnaíle, pers. comm., 03/07/2015.

[14] `http://multidict.net/`.

[15] Available at `https://www2.smo.uhi.ac.uk/gaidhlig/faclair/bunadas/`.

[16] `http://www.smo.uhi.ac.uk/en/rannsachadh/rnag2016/bunadas/`.

[17] `http://www.smo.uhi.ac.uk/en/rannsachadh/rnag2016/bunadas/`.

A.1.1.5 *Foclóir Gaedhilge agus Béarla* (Dinneen)

Fr. Patrick Stephen Dinneen's *Foclóir Gaedhilge agus Béarla* or Irish-English dictionary 'remains the most useful dictionary to scholars and readers of 18th- and 19th-century literature' (Ua Súilleabháin 2006: 588). The dictionary pre-dates the 1958 spelling and grammar standardisation and is printed in the *Cló Gaelach* (Irish typeface). The first edition dates from 1904, but a much-extended version appeared in 1927. The dictionary was the subject of various digitisation projects, including Irish-English Dictionary online and the unfinished Digital Dinneen (cf. below under Section A.2).

A.1.1.5.1 PDF

A PDF version of the first edition of Dinneen (1904) has been prepared by Alan Mac an Bhaird, available at Corpus of Electronic Texts (CELT), cf. section A.1.2.1.[18]

A.1.1.5.2 Irish-English Dictionary online

A project of the University of Limerick, this resource offers a digitised dictionary based on a scanned version of Dinneen's 1927 edition.[19] Functionalities include online browsing through the dictionary and searching by English and by Irish words. Although the search tool only allows pre-standard orthography, it recognises parts of words and certain wildcards. Pointers are created in the form of hyperlinks that direct the reader to the relevant scanned page of Dinneen where the headword is found.

A.1.1.6 *Foclóir Stairiúil na Gaeilge*

Background and purpose — *Foclóir Stairiúil na Gaeilge*,[20] 'the Historical Dictionary of Irish', was established in the Royal Irish Academy in 1976 with Tomás de Bhaldraithe[21] as general editor. This dictionary project builds on the Academy's *Dictionary of the Irish Language Based Mainly on Old and Middle Irish Materials* (Quin 1983), now digitised as eDIL, cf. section A.1.1.1), which covers the period up to c. 1650.[22] It was envisaged that *Foclóir Stairiúil na Gaeilge* should use the modern form as a headword with the history of words being traced back to the beginning of the 17th century (*Corpas na Gaeilge 1600–1882* (2004)). As Uí Dhonnchadha et al. (2014) have stated, challenges for the compilers are daunting: the Irish language is in decline from 1650 onwards and dialects come to the fore, with written dialects replacing a standard literary language.

Contents and mark-up — the dictionary entries will be drafted from a digitised corpus consisting of about 90+ million words, it is estimated (Uí Dhonnchadha et al. 2014: 13). Headwords will be extracted from *Corpas Stairiúil na Gaeilge* (cf. section A.1.2.4).

[18] `https://celt.ucc.ie//Dinneen1sted.html`.

[19] `http://glg.csisdmz.ul.ie/index.php`.

[20] `https://www.ria.ie/research-projects/focloir-stairiuil-na-gaeilge`.

[21] Cf. `http://www.ainm.ie/Bio.aspx?ID=1534` for a short biography (in Irish) of Tomás de Bhaldraithe.

[22] `https://www.ria.ie/aidhmeanna`.

Accessibility and level of search available — Natural Language Processing activities are employed on the basis of *Corpas Stairiúil na Gaeilge*, with the following goals (Uí Dhonnchadha et al. 2014):

a) search the corpus using modern spelling and find examples with earlier spelling;

b) avail of the Part-Of-Speech (POS) Tagger and lemmatisation tools available for modern orthography (post-1958) (Uí Dhonnchadha & van Genabith 2006).

The Dictionary's editiorial committee is currently modelling sample historical entries for the dictionary.[23]

A.1.2 Corpora

A.1.2.1 Corpus of Electronic Texts (CELT)

Background and purpose — the Corpus of Electronic Texts (CELT) at University College Cork, established in 1997, is Ireland's longest running Humanities Computing[24] project.[25] The project envisages a wide audience such as scholars, students, teachers and researchers interested in contemporary and historical topics from many areas, including literature and the other arts. The texts can be searched, read on-screen, downloaded for later use, or printed out.

Contents and mark-up — CELT has a searchable online textbase consisting of over 18.5 million words, currently representing 1621 texts and translations in various historical and contemporary European languages; there are 686 Irish (or Scottish Gaelic) source texts,[26] covering the following genres: Early Irish poetry, Irish Bardic Poetry, History, Law, Genealogy, Christian writings, Narrative, Irish originals, Poetry, Grammar, Metrics, Lexicology, Science & Medicine and Ecclesiology. Every historical period is represented, although the bulk of texts belong to the Early Irish and Early Modern Irish period. Categorisation is by genre, not by date, unless a genre is intrinsically connected to a language period, as with Early Irish Lyric Poetry and Bardic Poetry (the latter being in Classical Modern Irish).

Each text is accompanied by marked-up background details and bibliographic information, including dating and language of the text. Texts are marked-up with structural and analytic features according to the Text Encoding Initiative (TEI) standards.[27] Structural mark-up is employed for features such as stanzas, line and page breaks, while analytic mark-up accompanies personal and group names, place names and special terms (all shown in bold in the HTML), as well as diacritics and other editorial features.

Linguistic annotation — none.

Accessibility and level of search available — conversions to HTML are made for online reading, and the master files can be used to create versions in other formats, and for contextual

[23] https://www.ria.ie/obair-reatha.

[24] This field and its history is discussed in section 1.1.

[25] http://www.ucc.ie/celt.

[26] https://celt.ucc.ie/faq.html.

[27] http://www.tei-c.org/.

searching, concordancing, and other analyses. Texts are also available in eXtensible Mark-up Language (XML) and Standard Generalized Markup Language (SGML) format.

A.1.2.2 *Thesaurus Linguae Hibernicae* (TLH)

Background and purpose — *Thesaurus Linguae Hibernicae* (TLH) (2006-11)[28] was a project of the School of Irish, Celtic Studies, Irish Folklore & Linguistics at University College Dublin. According to the TLH website, '[I]t aims to provide web access to digital editions of texts in Early and Medieval Irish as a research tool for scholars and resource for teachers'.[29] The project follows the guidelines of the Text Encoding Initiative (TEI)[30] for digital scholarly editions and aims to provide digital editions of the following materials (in eXtensible Markup Language (XML)):

- Texts in the Franciscan A manuscripts (11th–17th centuries), now in the custody of University College Dublin
- New diplomatic transcriptions of published and unpublished texts.
- Scholarly editions no longer easily available

Contents and mark-up — TLH incorporates 223 Early and Medieval Irish texts, approximating 300,000 words, encoded in XML. Texts are accompanied by a header file and translation. The header file gives bibliographic information and date; however, only the language period is given, no detailed dating is provided. Mark-up is present to represent editorial features.

Linguistic annotation — none.

Accessibility and level of search available — publicly accessible. A search facility for both the Irish texts and the translations are available, giving the option to search the XML encoded files or all TLH.

A.1.2.3 Bardic Poetry corpus

Background and purpose — the Bardic Poetry corpus consists of bardic material composed mainly during the Classical Modern Irish period (13th to the 17th century), with some predating the 13th century and some post-dating this period, i.e., belonging to the early 18th century. In the context of the Higher Education Authority-funded Bardic Poetry project[31] in the Irish Department of Trinity College (2000–2006), and in collaboration with Dr Katharine Simms

[28] `http://www.ucd.ie/tlh/`.

[29] `http://www.ucd.ie/tlh/about.html`.

[30] `http://www.tei-c.org/`.

[31] Part of PRTLI Cycle I and III, 2000–2003 and 2002-2006, respectively. For more information on this project cf. `https://www.tcd.ie/CISS/bardic.php`.

of the History Department in Trinity College Dublin, 650 previously unpublished Bardic poems in Irish and British libraries were transcribed, 500 of which appeared in McManus & Ó Raghallaigh (2010).

Contents and mark-up — Dr Katharine Simms had been working on a database of bardic poems (published and unpublished) for many years previous to the above-mentioned Bardic Poetry project. This database is available online with various search options to facilitate philological and historical research (one can bring up patrons associated with poems etc.).[32] It is currently being updated and expanded by Dr Mícheál Hoyne, School of Celtic Studies, DIAS. This collection of 650 poems was added to a corpus of 1,400 previously published poems, which together constitute the Bardic Poetry corpus. The metadata (date of composition, patron, etc.) is incorporated for each text in the corpus.

Linguistic annotation — the corpus has been tagged by the tagger for Modern Irish (Uí Dhonnchadha & van Genabith 2006), which is in the stage of being adapted for *Corpas Stairiúil na Gaeilge* (Uí Dhonnchadha et al. 2014), cf. section A.1.2.4, showing promising results.[33]

Accessibility and level of search available — the corpus has been loaded into the corpus query tool Sketch Engine[34], which facilitates various corpus-search functionalities, but it is not (yet) publicly available.[35]

A.1.2.4 *Corpas Stairiúil na Gaeilge*

Background and purpose — in the context of the Royal Irish Academy's ongoing unilingual historical Modern Irish dictionary project *Foclóir Stairiúil na Gaeilge* (cf. section A.1.1.6), covering the period 1600–2000, various texts have been digitised and published. Sample headword entries are currently extracted from this corpus.

Contents and mark-up — various publications and text archives constitute the corpus. The pre-20th and early 20th century material has been published online,[36] constituting two corpora:

1. *Corpas (Stairiúil) na Gaeilge* 1 (1600–1882), initially published on CD-ROM in 2004 (7.25 million words).

2. *Corpas Stairiúil na Gaeilge* 2 (1882-1926), texts published by *Connradh na Gaedhilge* (modern *Conradh na Gaeilge*) (also approximately 7 million words).

In addition to this, a digital Corpus of the Gaelic Journal (1882-1909) has been prepared

[32] Available at `https://bardic.celt.dias.ie`.

[33] Dr Eoin Mac Cárthaigh, presentation as part of the Bardic Poetry Workshop, held on 12/05/2017 at Trinity College Dublin, cf. `https://bardicpoetryworkshop.wordpress.com`.

[34] `https://www.sketchengine.eu/`.

[35] Dr Eoin Mac Cárthaigh has presented on some of the search functionalities, cf. footnote 33. Cf. also Mac Cárthaigh (2018).

[36] `http://corpas.ria.ie/`.

and has been integrated into *Corpas Stairiúil na Gaeilge* 2, but is also accessible as a standalone corpus.[37]

Linguistic annotation — words in pre-standard orthography in *Corpas Stairiúil na Gaeilge* 1600–1926 are modernised using a standardiser (Scannell 2017) and processed with a Part-Of-Speech Tagger lemmatisation tools for standardised Modern Irish (Uí Dhonnchadha & van Genabith 2006). The web-based corpus query system Sketch Engine[38] is used to facilitate the retrieval of historical variants using a modern-language lemma.

Accessibility and level of search available — *Corpas Stairiúil na Gaeilge* 1600–1926 is available online and accompanied by a search interface that allows one to search for headword, standardised word or exact match, together with Part-Of-Speech. A successful query generates a list of variants, the text(s) that the word occurs in, and its context. Clicking on the word in context brings one to the text, with the instances of the queried word highlighted. The text is downloadable in four different formats: a user-friendly on-screen version, an eXtensible Markup Language (XML) version compliant with TEI,[39] the raw, unformatted text and an ePub version.

A.2 Unavailable

A.2.1 Miscellaneous

A.2.1.1 Electronic Lexicon of Medieval Irish

Background and purpose — the Electronic Lexicon of Medieval Irish[40] accompanies Nyhan (2006a), a Ph.D. thesis in which the author sets out to remedy the limitations in the hard-copy version of Dictionary of the Irish language (cf. section A.1.1.1). Nyhan's main research question was how a retro-digitised electronic dictionary (DIL) could be restructured using eXtensible Markup Language (XML), in turn facilitating support a deeper level of inquiry, i.e., to identify and return inflected medieval forms with a high degree of precision.

Contents and mark-up — the Electronic Lexicon of Medieval Irish (Lexicon) consists of a digitised subset of DIL, encoded and restructured in XML. Code Example A.1 and Code Example A.2 show XML snippets from the entry *téit* 'goes' in eDIL and in Nyhan (2006a) respectively, showing the difference in mark-up. While inflected forms in the Lexicon are embedded in a hierarchical grammatical structure, forms in eDIL are interspersed with references, disassociated from their grammatical tags and invariably tagged `oVar` (orthographical variant). It should be noted, however, that restructuring the dictionary has never been the aim of the eDIL project. As stated on the website, '[w]e have not attempted to iron out inconsistencies

[37] `http://irisleabharnagaedhilge.fng.ie/`.

[38] `https://www.sketchengine.eu/`.

[39] Text Encoding Initiative, cf. `http://www.tei-c.org/`.

[40] Referred to as Electronic Lexicon of Old Irish in Nyhan (2006a) and as Lexicon of Medieval Irish on the designated page on the CELT website (`http://www.ucc.ie/celt/digineen.html`). As this lexicon is based on eDIL, which is sourced mainly from Early Irish material, I adhere to the term Lexicon of Medieval Irish.

in the original Dictionary: our aim has been to use the time available to add new information rather than reorder existing material'.[41] In the current edition of eDIL, however, annotated inflected forms have been extracted and are presented below the headword.

Nyhan (2006a: 252–256) reports on possibilities for interlinking the Lexicon and CELT on the word/phrase level by creating arbitrary, user-generated links using `Javascript` code (Nyhan 2006a: 254–255, Nyhan 2006b: 153, Nyhan 2008: 9-10). The screenshot in Figure A.1, taken from Nyhan (2006a: 255), shows a web interface (unavailable) through which a user can highlight a word in a text on CELT and look it up in the Lexicon.

Accessibility and level of search available — Nyhan's research was done in close conjunction with CELT and the Electronic Publishing Unit, both at University College, Cork. Unfortunately, hyperlinks to a prototype of this electronic Lexicon found on various CELT webpages are all broken, and the resource, apparently, was never fully published.[42]

Code Example A.1 – Snippet of eDIL's XML code for entry *téit* 'goes'.

```
<entry>
<form><orth xml:id="1 téit">1 téit</orth></form>
<form><p>(see <bibl><title target="Ériu"
...
Irreg. <pos>vb.</pos>
with forms from <br column="124" line="59"/>
various roots.</p></form>
...
<form><p><br column="124" line="61"/><b>A</b>.
Early forms.</p></form>
<form><p>
<br column="124" line="62"/><mood>Indic.</mood>
<tns>pres.</tns>
...
<per>3</per> <number>s.</number> <oVar>téit</oVar>,
<bibl><title target="Ml" xml:id="d0e161066">Ml</title>
...
<bibl><orphanScope target="d0e161066">109<sup>a</sup>2
</orphanScope></bibl>. <oVar>teit</oVar>,
...
</p></form>
...
</entry>
```

[41] `http://dil.ie/about`.

[42] Peter Flynn, former head of Academic & Collaborative Technologies Unit in the University College Cork IT Services and close collaborator of Nyhan in the past, has informed me that 'there remain some decisions about formatting, presentation, and functionality which need to be taken first' (pers. comm., 04/11/2014). The following link to a sample of the Lexicon was kindly provided to me by Mr. Flynn: `http://research.ucc.ie/lexicon/sample`. This is a sample of the letter B and does not contain any verb lemmas.

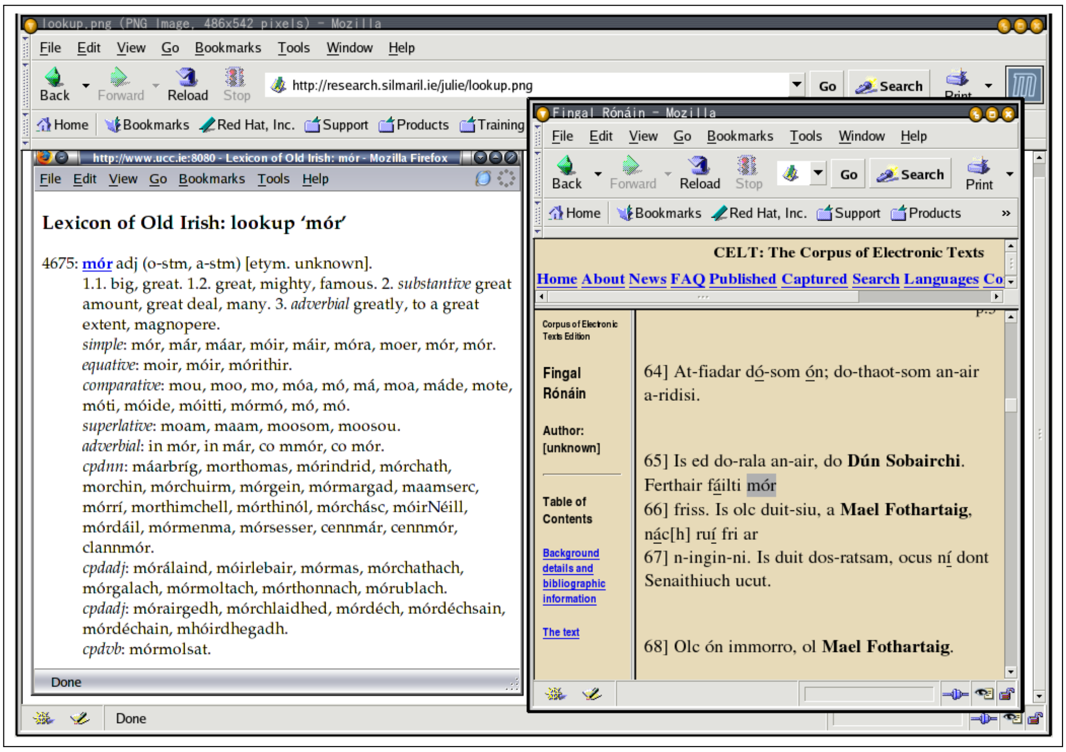
Code Example A.2 – Snippet of XML code for the restructured DIL entry *téit* ‘goes’ (Nyhan 2006a: 195–239).

```
<entry id="34789">
<lemma htype="1">téit</lemma>
 <gramgrp pos="vb">
  <itype></itype>
 </gramgrp>
 <paradigm>
  <mood type="indicative">
   <tense type="present">
    <number type="sg">
    ...
     <person n="3">
      <form type="regular">téit</form>
      <form type="regular">tét</form>
      <form type="regular">-tét</form>
      <form type="regular">-tet</form>
      <form type="regular">-téd</form>
      <form type="regular">-téd</form>
      <form type="regular">-téit</form>
      <form type="regular">tiat</form>
      <form type="regular">-tiat</form>
      <form type="regular">-téige</form>
      <form type="regular">teit</form>
      <form type="regular">téd</form>
      ...
      <form type="with-suffix-pron">téte</form>
      <form type="with-suffix-pron">téite</form>
      ...
     </person>
     ...
    </number>
    ...
   </tense>
   ...
  </mood>
  ...
 </paradigm>
</entry>
```

A.2.1.2 Linking Dictionaries and Texts

Background and purpose — the Linking Dictionaries and Texts (LDT) project,[43] funded by the Irish Higher Education Authority, was a North-South Ireland collaboration between the University of Ulster, Coleraine, and University College Cork (Nyhan 2008). The goal of the project was to create interoperability between the electronic Dictionary of the Irish language

[43] http://www.ucc.ie/celt/LDT.html.


lookup.png (PNG Image, 486x542 pixels) - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://research.silmaril.ie/julie/lookup.png
Go Search Print
Home Bookmarks Red Hat, Inc. Support Products Training
http://www.ucc.ie:8080 - Lexicon of Old Irish: mór - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Done
Done
Fingal Rónáin - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop Go Search Print
Home Bookmarks Red Hat, Inc. Support Products »

Lexicon of Old Irish: lookup 'mór'

4675: mór adj (o-stm, a-stm) [etym. unknown].
1.1. big, great. 1.2. great, mighty, famous. 2. *substantive* great amount, great deal, many. 3. *adverbial* greatly, to a great extent, magnopere.
simple: mór, már, máar, móir, máir, móra, moer, mór, mór.
equative: moir, móir, mórithir.
comparative: mou, moo, mo, móa, mó, má, moa, máde, mote, móti, móide, móitti, mórmó, mó, mó.
superlative: moam, maam, moosom, moosou.
adverbial: in mór, in már, co mmór, co mór.
cpdnn: máarbríg, morthomas, mórindrid, mórchath, morchin, mórchuirm, mórgein, mórmargad, maamserc, mórrí, morthimchell, mórthinól, mórchásc, móirNéill, mórdáil, mórmenma, mórsesser, cennmár, cennmór, clannmór.
cpdadj: mórálaind, móirlebair, mórmas, mórchathach, mórgalach, mórmoltach, mórthonnach, mórublach.
cpdadj: mórairgedh, mórchlaidhed, mórdéch, mórdéchsain, mórdéchain, mhóirdhegadh.
cpdvb: mórmolsat.

CELT: The Corpus of Electronic Texts

Home About News FAQ Published Captured Search Languages Co

Corpus of Electronic Texts Edition

Fingal Rónáin

Author: [unknown]

Table of Contents

Background details and bibliographic information

The text

p.5

64] At-fiadar dó-som ón; do-thaot-som an-air a-ridisi.

65] Is ed do-rala an-air, do **Dún Sobairchi**. Ferthair fáilti mór
66] friss. Is olc duit-siu, a **Mael Fothartaig**, nác[h] ruí fri ar
67] n-ingin-ni. Is duit dos-ratsam, ocus ní dont Senaithiuch ucut.

68] Olc ón immorro, ol **Mael Fothartaig**.

Figure A.1 – Web interface linking CELT (cf. section A.1.2.1) and the Lexicon of Medieval Irish on the word level with a lookup mechanism (Nyhan 2006a: 255). This tool is not available.

(eDIL) (cf. section A.1.1.1) and Corpus of Electronic Texts (CELT, cf. section A.1.2.1), as well as creating electronic editions of the most commonly cited texts in DIL, to facilitate researchers in that they will be able to retrieve the text and its context from a word in eDIL on their own PC. The envisaged interlinking of dictionaries and texts is illustrated in Figure 1.2 in section 1.5 in the main part of this thesis.

Contents and mark-up — according to the dedicated webpage on CELT, Julianne Nyhan designed a program to automate the creation of remote, fixed links between bibliographical citations in eDIL and texts on CELT, further explained in Nyhan (2006a: 258–259):

> It was envisaged that this cooperation would take the form of conventional predetermined (fixed) links, either encoded in a static HTML file or using HTML generated from an SGML or XML source. This would enable users to click on a bibliographical citation (in the eDIL or to a lesser extent the Lexicon) that would resolve to the specified text in the CELT website.

Furthermore, Nyhan (2006b) mentions research carried out at CELT into an eXtensible Stylesheet Language Transformations[44] (XSLT) lookup tool to facilitate the links.

To maximise usefulness of automated links between eDIL and CELT, 2 million words of XML encoded Irish texts from c. 800–1650 have been added to CELT—especially ones that

[44] A language for transforming XML into other XML documents or other formats, visualised and explained at `https://www.w3.org/standards/xml/transformation`.

are frequently cited in eDIL. According to the website, this target was reached by September 2006.

Accessibility and level of search available — according to the website, the generation of automated links has been delayed.

A.2.1.3 Digital Dinneen

Background and purpose — as stated on the CELT website, the Electronic Lexicon of Medieval Irish (cf. section A.2.1.1) is the background to the Digital Dinneen project.[45] The project ran from 2005-2008. Its aim was to complement digital tools already in place for older stages of the language, facilitating an understanding of the diachronic development of the Irish language. Digital Dinneen is envisaged as an integrated resource, incorporated into and interoperable with the Electronic Lexicon of Medieval Irish and with Corpus of Electronic Texts (CELT, cf. section A.1.2.1).

Contents mark-up — a digitised, XML-encoded edition of Dinneen's *Foclóir Gaedhilge agus Béarla* (cf. section A.1.1.5). Pointers generated from the headwords 'allow a user to follow a modern Irish form in Dinneen's dictionary back to its earlier forms in eDIL and the [electronic] Lexicon [of Medieval Irish]' (Nyhan 2008: 6). Instrumental in this linking is the use of de Bhaldraithe (1981), which provides an alphabetical index of Modern Irish words accompanied by their corresponding entries in the Dictionary of the Irish language, or DIL (cf. section A.1.1.1).

To alleviate the 'problem' with the older typescript and pre-standard orthography in *Foclóir Gaedhilge agus Béarla*, Nyhan (2008) puts forward the idea of incorporating the post-spelling reform orthography in the headword meta-data of Digital Dinneen.

As an integrated edition, it is envisaged that the linking technology on the word/phrase and citational/textual level, developed in the context of the Electronic Lexicon of Medieval Irish and the Linking Dictionaries and Texts project (cf. section A.2.1.2), respectively, will be extended to Digital Dinneen (Nyhan 2006b, 2008). According to its website, CELT's textbase will be expanded accordingly with Irish texts from the 17th–20th centuries.

Accessibility and level of search available — the resource is not available.[46]

[45] For information on this project cf. `http://www.ucc.ie/celt/digineen.html`. The idea for this project stems from Beatrix Färber (pers. comm. 03/11/2014).

[46] The XML files are at CELT but there are no tools involved (Beatrix Färber, pers. comm. 30/10/2014) and a lookup mechanism or search interface has not been implemented (Julianne Nyhan, pers. comm., 23/02/2012).

Appendix B

Old Irish verb paradigms

The paradigms on pages 162–174 are taken from Green (1995), who employs italics to denote unattested inflections that are reconstructed from attested forms (but this distinction is not made for the sample weak verbs, of which *marbaid* and *léicid* are included here). Green (1995) uses the present stem classification system from Thurneysen (1946). I will give the alternative classification (W(eak), S(trong)) from McCone (1997) below as well. The abbreviation v.n. stands for verbal noun.

beirid (page 162): BI / S1a
Suppletive stem:
ro·ucca (page 163): AI / W1.[1]

crenaid (pages 164–165): BIV / S3.

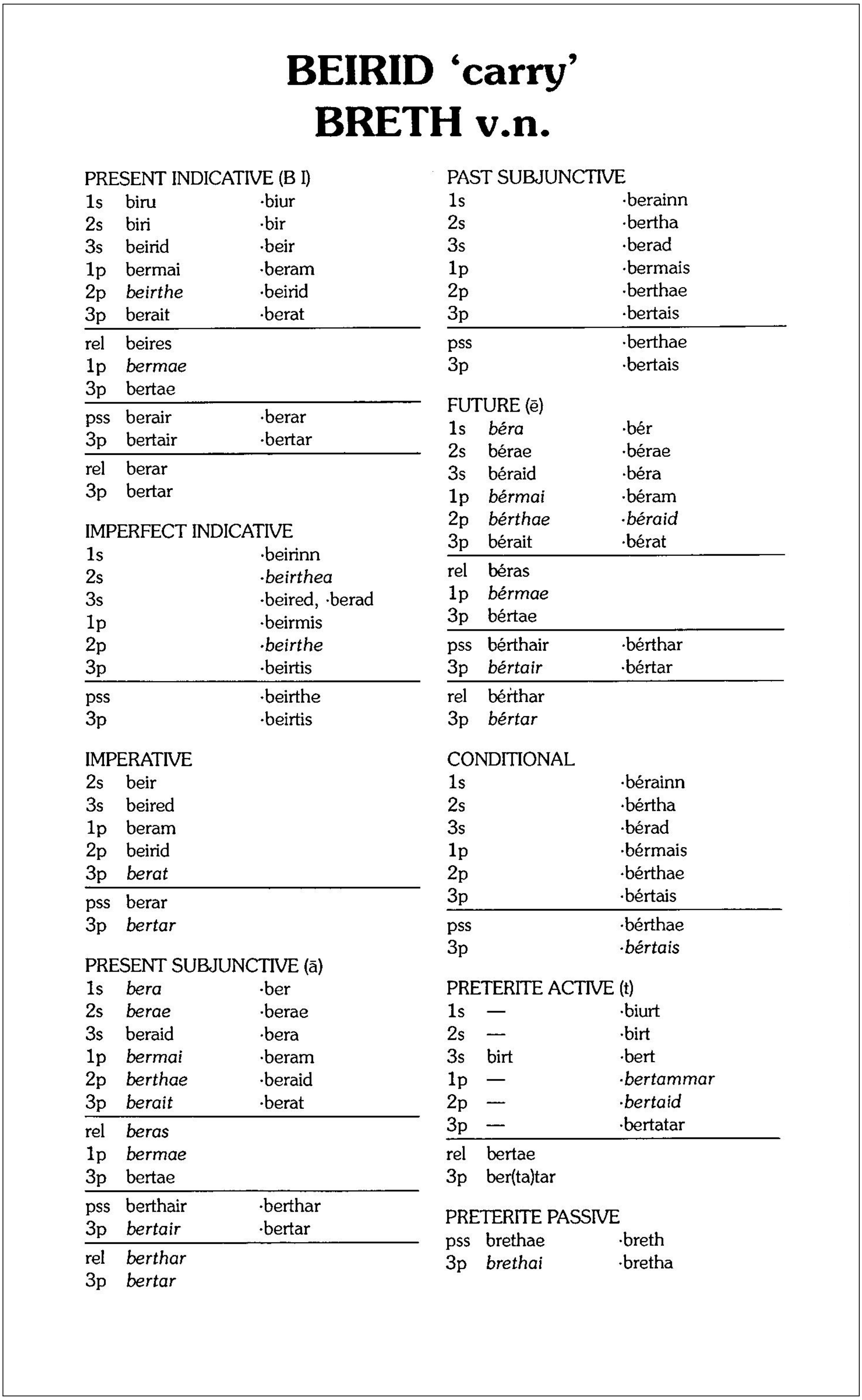
do·beir (pages 166–167): BI / S1a.
Suppletive stems:
do·ratai (page 168): AII / W2a.
do·uccai (page 169): AII / W2b.

gaibid (pages 170–171): BII / S2.

léicid (page 172): AII / W2a.

marbaid (pages 173–174): AI / W1.

[1] Stifter (2006: 374) gives *ro·uccai*, and accordingly present stem class W2b (AII).



BEIRID ‘carry’
BRETH v.n.

PRESENT INDICATIVE (B I)

1s	biru	·biur
2s	biri	·bir
3s	beirid	·beir
1p	bermai	·beram
2p	*beirthe*	·beirid
3p	berait	·berat
rel	beires	
1p	*bermae*	
3p	bertae	
pss	berair	·berar
3p	bertair	·bertar
rel	berar	
3p	bertar	

IMPERFECT INDICATIVE

1s		·beirinn
2s		*·beirthea*
3s		·beired, ·berad
1p		·beirmis
2p		*·beirthe*
3p		·beirtis
pss		·beirthe
3p		·beirtis

IMPERATIVE

2s	beir	
3s	beired	
1p	beram	
2p	beirid	
3p	*berat*	
pss	berar	
3p	*bertar*	

PRESENT SUBJUNCTIVE (ā)

1s	*bera*	·ber
2s	*berae*	·berae
3s	beraid	·bera
1p	*bermai*	·beram
2p	*berthae*	·beraid
3p	*berait*	·berat
rel	*beras*	
1p	*bermae*	
3p	bertae	
pss	berthair	·berthar
3p	*bertair*	·bertar
rel	*berthar*	
3p	*bertar*	

PAST SUBJUNCTIVE

1s		·berainn
2s		·bertha
3s		·berad
1p		·bermais
2p		·berthae
3p		·bertais
pss		·berthae
3p		·bertais

FUTURE (ē)

1s	*béra*	·bér
2s	bérae	·bérae
3s	béraid	·béra
1p	*bérmai*	·béram
2p	*bérthae*	*·béraid*
3p	bérait	·bérat
rel	béras	
1p	*bérmae*	
3p	bértae	
pss	bérthair	·bérthar
3p	*bértair*	·bértar
rel	béŕthar	
3p	*bértar*	

CONDITIONAL

1s		·bérainn
2s		·bértha
3s		·bérad
1p		·bérmais
2p		·bérthae
3p		·bértais
pss		·bérthae
3p		*·bértais*

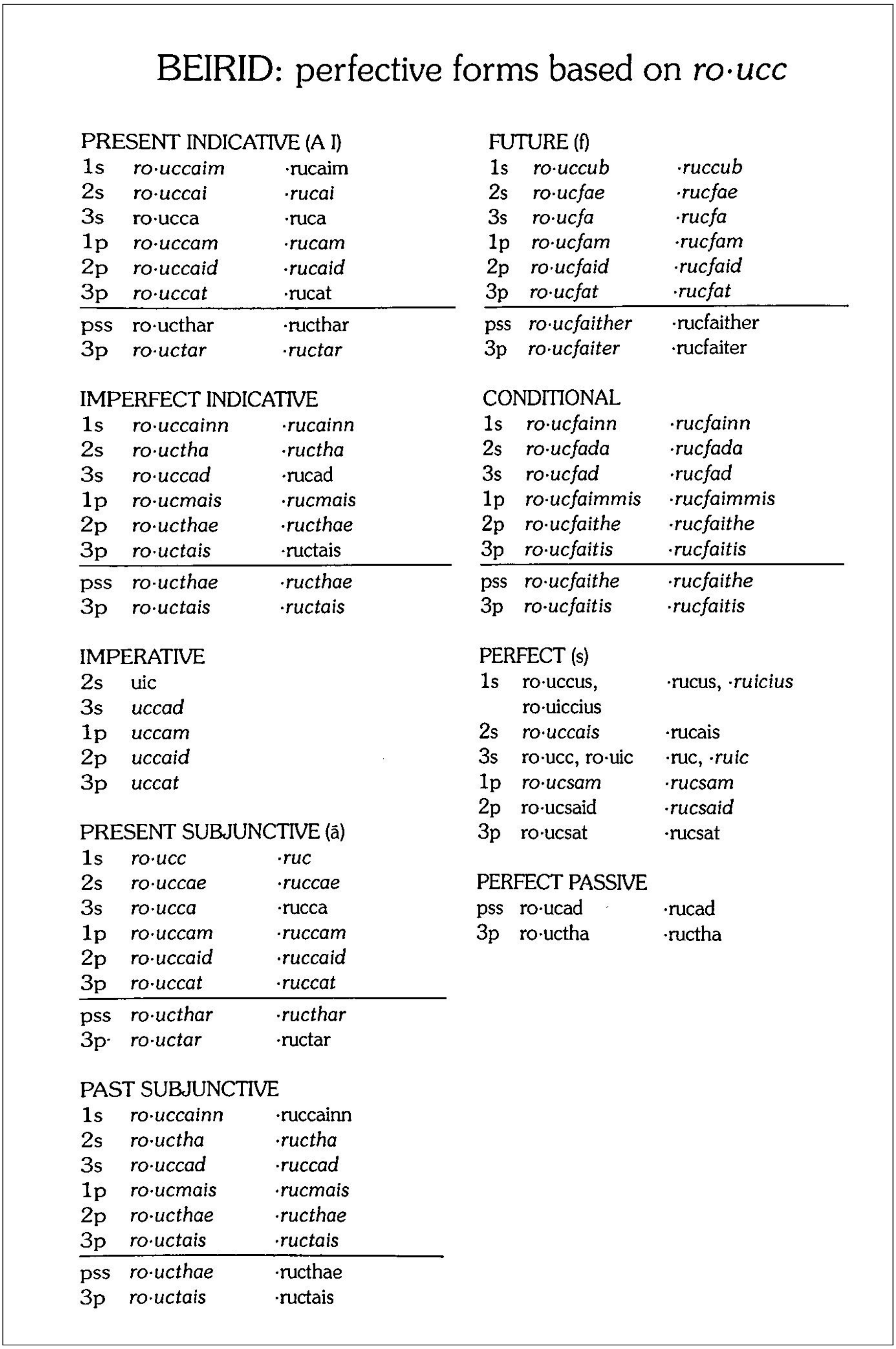
PRETERITE ACTIVE (t)

1s	—	·biurt
2s	—	·birt
3s	birt	·bert
1p	—	*·bertammar*
2p	—	*·bertaid*
3p	—	·bertatar
rel	bertae	
3p	ber(ta)tar	

PRETERITE PASSIVE

pss	brethae	·breth
3p	*brethai*	·bretha

Figure B.1 – Paradigm for *beirid* (S1a). Taken from Green (1995: 22).



BEIRID: perfective forms based on *ro·ucc*

PRESENT INDICATIVE (A I)		
1s	*ro·uccaim*	·rucaim
2s	*ro·uccai*	*·rucai*
3s	ro·ucca	·ruca
1p	*ro·uccam*	*·rucam*
2p	*ro·uccaid*	*·rucaid*
3p	*ro·uccat*	·rucat
pss	ro·ucthar	·ructhar
3p	*ro·uctar*	*·ructar*

IMPERFECT INDICATIVE		
1s	*ro·uccainn*	*·rucainn*
2s	*ro·uctha*	*·ructha*
3s	*ro·uccad*	·rucad
1p	*ro·ucmais*	*·rucmais*
2p	*ro·ucthae*	*·ructhae*
3p	*ro·uctais*	·ructais
pss	*ro·ucthae*	*·ructhae*
3p	*ro·uctais*	*·ructais*

IMPERATIVE	
2s	uic
3s	*uccad*
1p	*uccam*
2p	*uccaid*
3p	*uccat*

PRESENT SUBJUNCTIVE (ā)		
1s	*ro·ucc*	*·ruc*
2s	*ro·uccae*	*·ruccae*
3s	*ro·ucca*	·rucca
1p	*ro·uccam*	*·ruccam*
2p	*ro·uccaid*	*·ruccaid*
3p	*ro·uccat*	*·ruccat*
pss	*ro·ucthar*	*·ructhar*
3p·	*ro·uctar*	·ructar

PAST SUBJUNCTIVE		
1s	*ro·uccainn*	·ruccainn
2s	*ro·uctha*	*·ructha*
3s	*ro·uccad*	*·ruccad*
1p	*ro·ucmais*	*·rucmais*
2p	*ro·ucthae*	*·ructhae*
3p	*ro·uctais*	*·ructais*
pss	*ro·ucthae*	·ructhae
3p	*ro·uctais*	·ructais

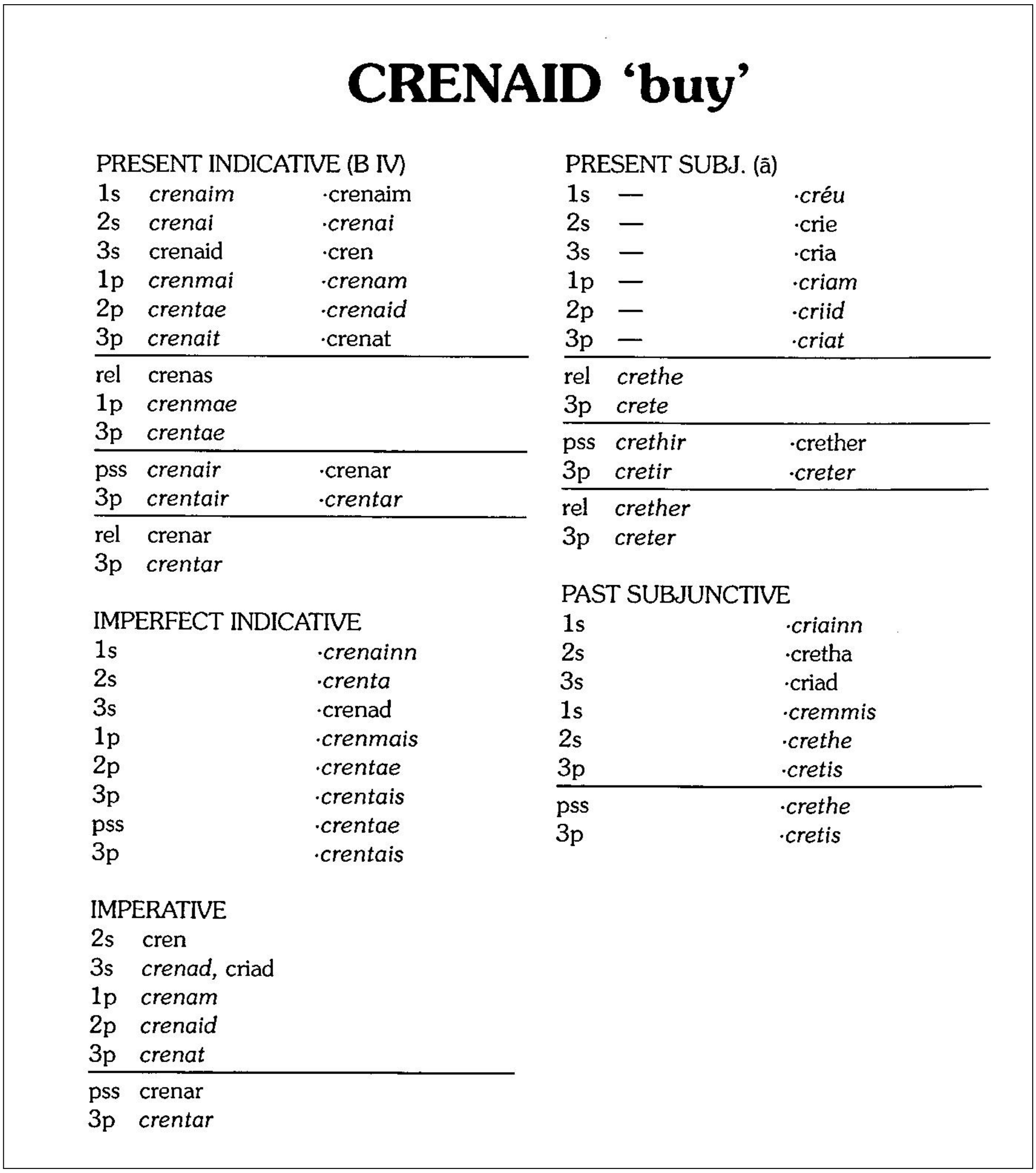
FUTURE (f)		
1s	*ro·uccub*	*·ruccub*
2s	*ro·ucfae*	*·rucfae*
3s	*ro·ucfa*	*·rucfa*
1p	*ro·ucfam*	*·rucfam*
2p	*ro·ucfaid*	*·rucfaid*
3p	*ro·ucfat*	*·rucfat*
pss	*ro·ucfaither*	·rucfaither
3p	*ro·ucfaiter*	·rucfaiter

CONDITIONAL		
1s	*ro·ucfainn*	*·rucfainn*
2s	*ro·ucfada*	*·rucfada*
3s	*ro·ucfad*	*·rucfad*
1p	*ro·ucfaimmis*	*·rucfaimmis*
2p	*ro·ucfaithe*	*·rucfaithe*
3p	*ro·ucfaitis*	*·rucfaitis*
pss	*ro·ucfaithe*	*·rucfaithe*
3p	*ro·ucfaitis*	*·rucfaitis*

PERFECT (s)		
1s	ro·uccus, ro·uiccius	·rucus, *·ruicius*
2s	*ro·uccais*	·rucais
3s	ro·ucc, ro·uic	·ruc, *·ruic*
1p	*ro·ucsam*	*·rucsam*
2p	ro·ucsaid	*·rucsaid*
3p	ro·ucsat	·rucsat

PERFECT PASSIVE		
pss	ro·ucad	·rucad
3p	ro·uctha	·ructha

Figure B.2 – Paradigm for *ro·ucca(i)* (W1/W2b, indep. perf. active 3sg. *ro·ucc*, *ro·uic*), suppletive stem of *beirid*. Taken from Green (1995: 23).



CRENAID 'buy'

PRESENT INDICATIVE (B IV)

1s	*crenaim*	·crenaim
2s	*crenai*	*·crenai*
3s	crenaid	·cren
1p	*crenmai*	*·crenam*
2p	*crentae*	*·crenaid*
3p	*crenait*	·crenat
rel	crenas	
1p	*crenmae*	
3p	*crentae*	
pss	*crenair*	·crenar
3p	*crentair*	*·crentar*
rel	crenar	
3p	*crentar*	

IMPERFECT INDICATIVE

1s	*·crenainn*
2s	*·crenta*
3s	·crenad
1p	*·crenmais*
2p	*·crentae*
3p	*·crentais*
pss	*·crentae*
3p	*·crentais*

IMPERATIVE

2s	cren
3s	*crenad,* criad
1p	*crenam*
2p	*crenaid*
3p	*crenat*
pss	crenar
3p	*crentar*

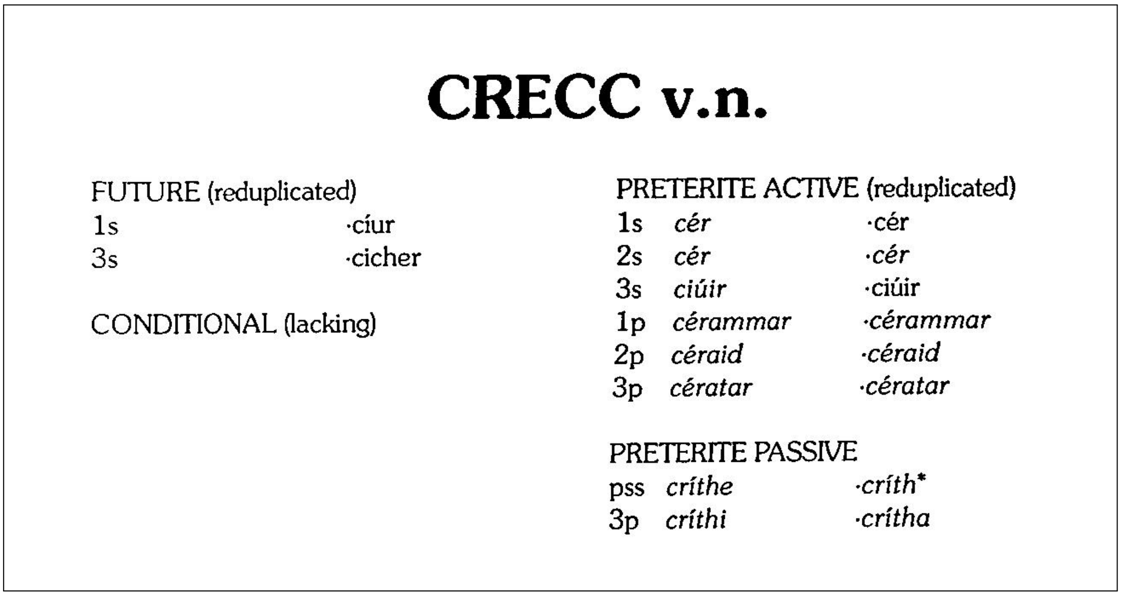
PRESENT SUBJ. (ā)

1s	—	*·créu*
2s	—	·crie
3s	—	·cria
1p	—	*·criam*
2p	—	*·criid*
3p	—	*·criat*
rel	*crethe*	
3p	*crete*	
pss	*crethir*	·crether
3p	*cretir*	*·creter*
rel	*crether*	
3p	*creter*	

PAST SUBJUNCTIVE

1s	*·criainn*
2s	·cretha
3s	·criad
1s	*·cremmis*
2s	*·crethe*
3p	*·cretis*
pss	*·crethe*
3p	*·cretis*

Figure B.3 – Paradigm for *crenaid* (S3). Taken from Green (1995: 34).



CRECC v.n.

FUTURE (reduplicated)

1s		·cíur
3s		·cicher

CONDITIONAL (lacking)

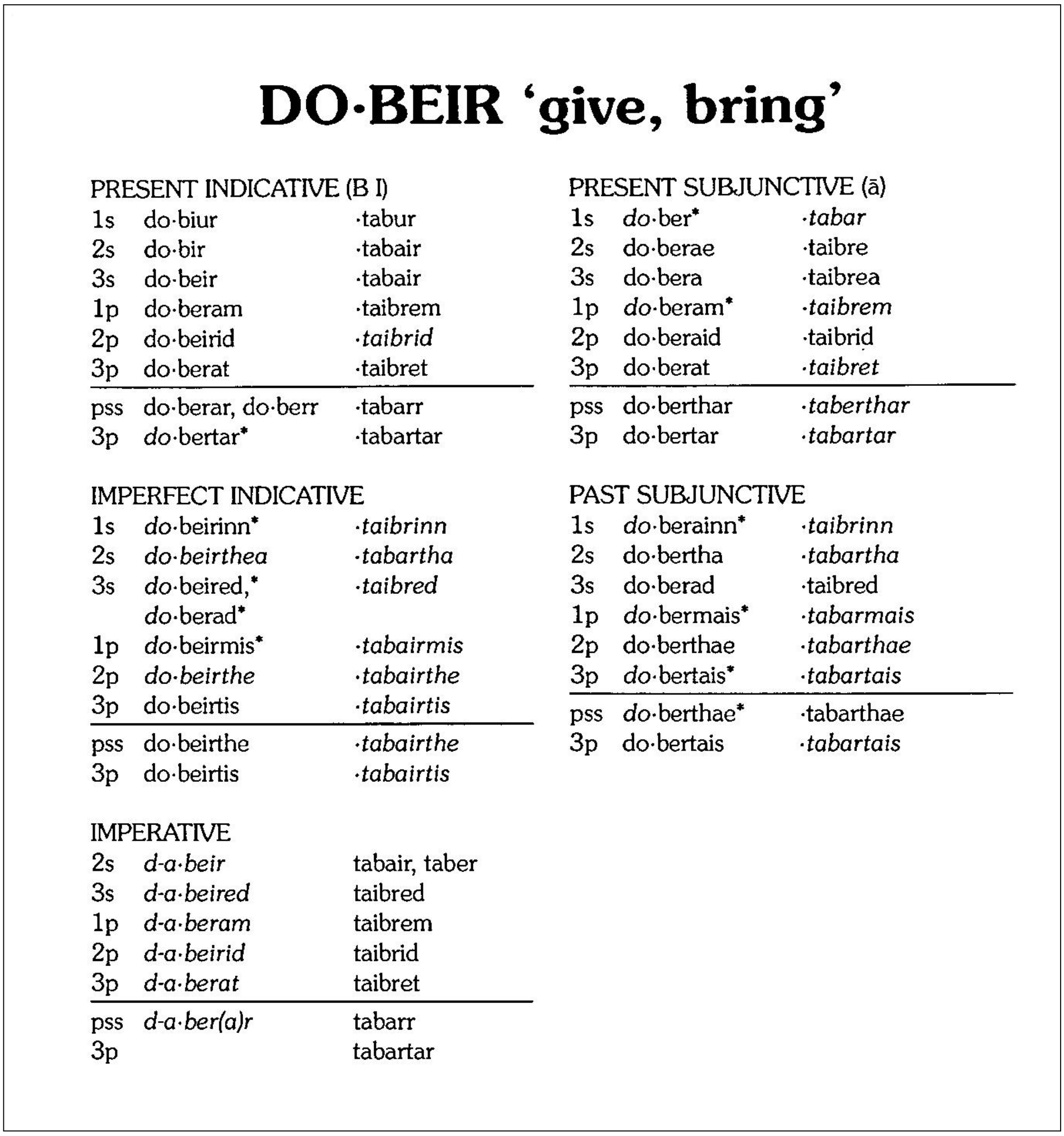
PRETERITE ACTIVE (reduplicated)

1s	*cér*	·cér
2s	*cér*	*·cér*
3s	*ciúir*	·ciúir
1p	*cérammar*	*·cérammar*
2p	*céraid*	*·céraid*
3p	*cératar*	*·cératar*

PRETERITE PASSIVE

pss	*críthe*	*·críth**
3p	*caríthi*	*·crítha*

Figure B.4 – Paradigm for *crenaid* (S3) (continued). Taken from Green (1995: 35). Green supplies the note '*According to GOI § 709. Not cited in VKG or Dict.' GOI = Thurneysen (1946), VKG = Pedersen (1909–13) and Dict. = (e)DIL.



DO·BEIR ‘give, bring’

PRESENT INDICATIVE (B I)

1s	do·biur	·tabur
2s	do·bir	·tabair
3s	do·beir	·tabair
1p	do·beram	·taibrem
2p	do·beirid	*·taibrid*
3p	do·berat	·taibret
pss	do·berar, do·berr	·tabarr
3p	*do*·bertar*	·tabartar

IMPERFECT INDICATIVE

1s	*do*·beirinn*	*·taibrinn*
2s	*do·beirthea*	*·tabartha*

3s	*do*·beired,*
	do·berad*
		·taibred

1p	*do*·beirmis*	*·tabairmis*
2p	*do·beirthe*	*·tabairthe*
3p	do·beirtis	*·tabairtis*
pss	do·beirthe	*·tabairthe*
3p	do·beirtis	*·tabairtis*

IMPERATIVE

2s	*d-a·beir*	tabair, taber
3s	*d-a·beired*	taibred
1p	*d-a·beram*	taibrem
2p	*d-a·beirid*	taibrid
3p	*d-a·berat*	taibret
pss	*d-a·ber(a)r*	tabarr
3p		tabartar

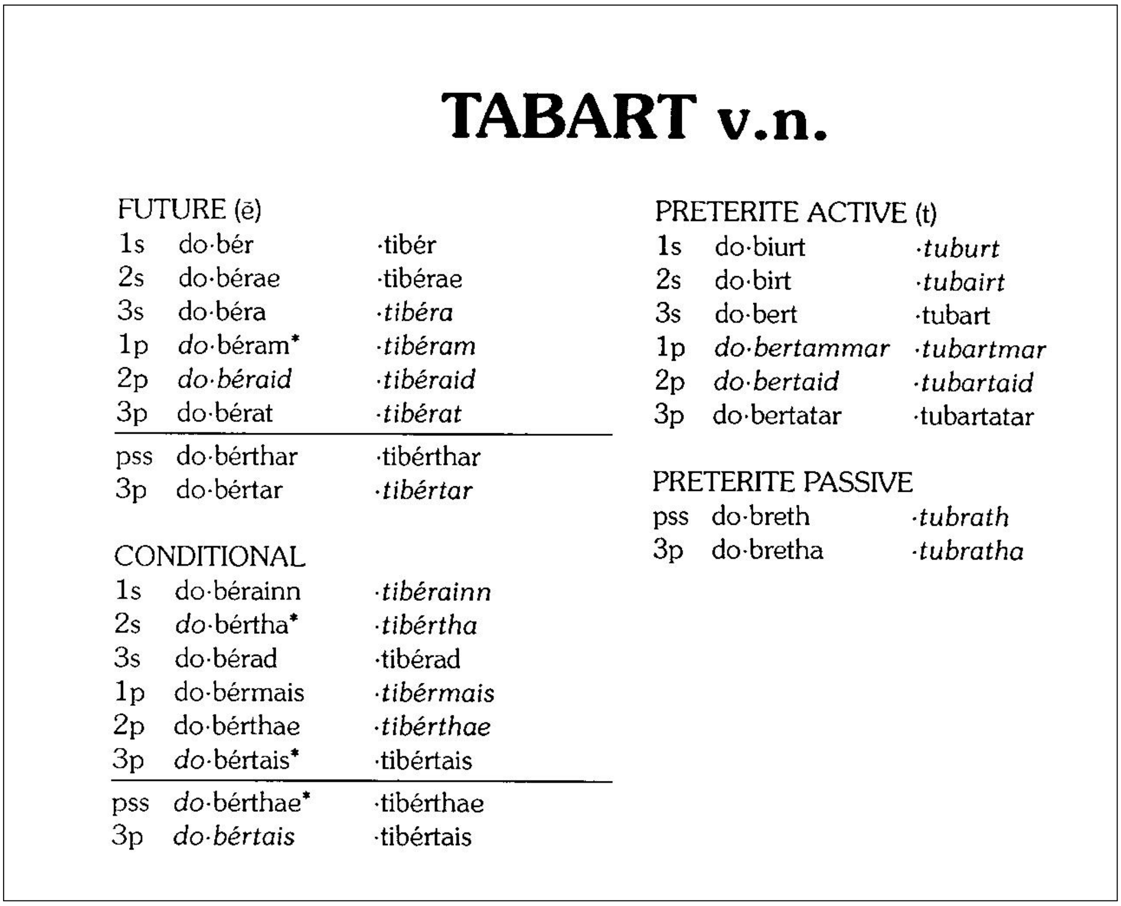
PRESENT SUBJUNCTIVE (ā)

1s	*do*·ber*	*·tabar*
2s	do·berae	·taibre
3s	do·bera	·taibrea
1p	*do*·beram*	*·taibrem*
2p	do·beraid	·taibrid
3p	do·berat	*·taibret*
pss	do·berthar	*·taberthar*
3p	do·bertar	*·tabartar*

PAST SUBJUNCTIVE

1s	*do*·berainn*	*·taibrinn*
2s	do·bertha	*·tabartha*
3s	do·berad	·taibred
1p	*do*·bermais*	*·tabarmais*
2p	do·berthae	*·tabarthae*
3p	*do*·bertais*	*·tabartais*
pss	*do*·berthae*	·tabarthae
3p	do·bertais	*·tabartais*

Figure B.5 – Paradigm for *do·beir* (S1a). Taken from Green (1995: 38), who adds that forms with * are attested for *beirid* or other compounds of *·beir*, e.g., *ar·beir*, *as·beir*, *con·beir*.



TABART v.n.

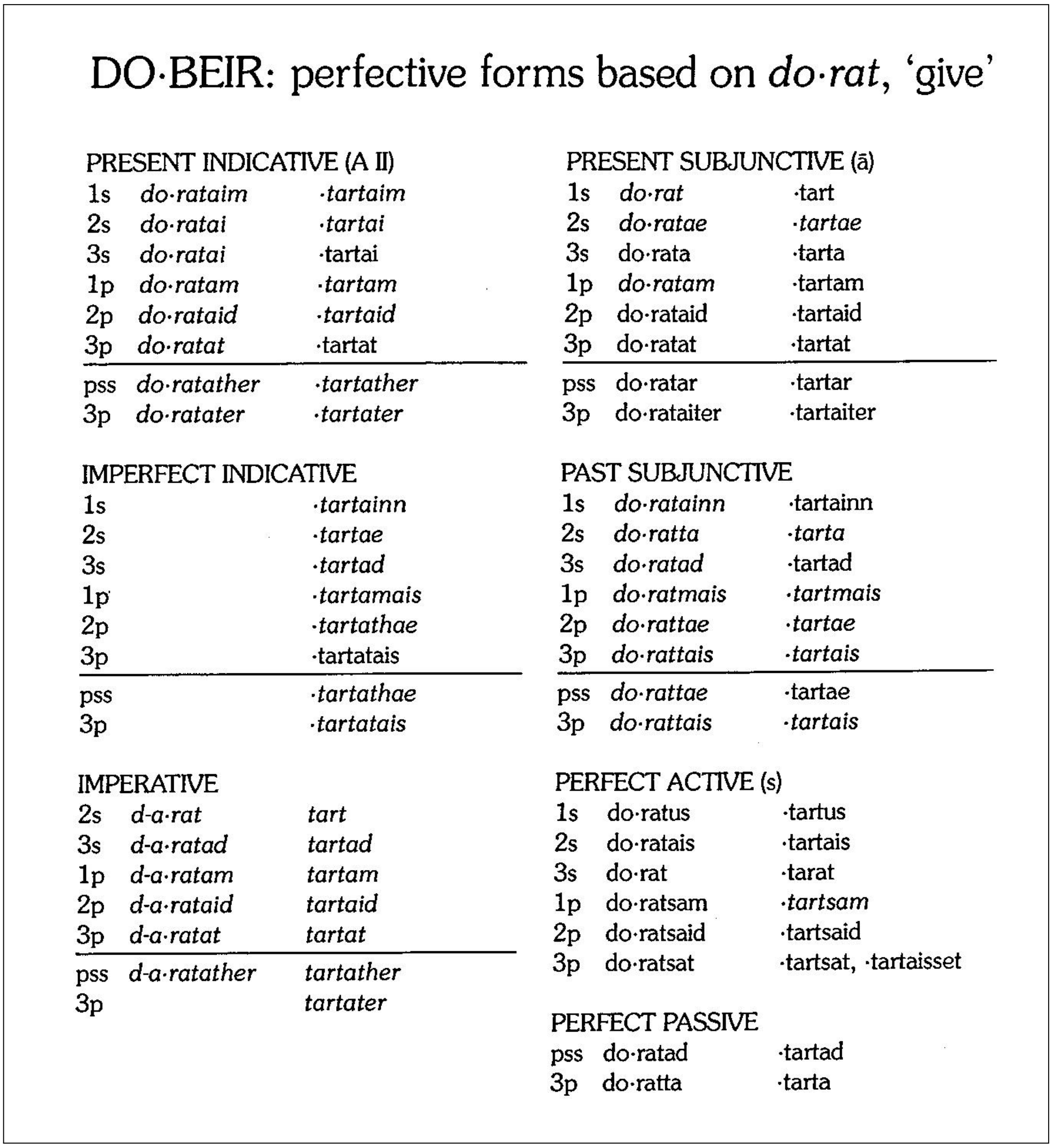
FUTURE (ē)		
1s	do·bér	·tibér
2s	do·bérae	·tibérae
3s	do·béra	*·tibéra*
1p	*do*·béram*	*·tibéram*
2p	*do·béraid*	*·tibéraid*
3p	do·bérat	*·tibérat*
pss	do·bérthar	·tibérthar
3p	do·bértar	*·tibértar*

CONDITIONAL		
1s	do·bérainn	*·tibérainn*
2s	*do*·bértha*	*·tibértha*
3s	do·bérad	·tibérad
1p	do·bérmais	*·tibérmais*
2p	do·bérthae	*·tibérthae*
3p	*do*·bértais*	·tibértais
pss	*do*·bérthae*	·tibérthae
3p	*do·bértais*	·tibértais

PRETERITE ACTIVE (t)		
1s	do·biurt	*·tuburt*
2s	do·birt	*·tubairt*
3s	do·bert	·tubart
1p	*do·bertammar*	*·tubartmar*
2p	*do·bertaid*	*·tubartaid*
3p	do·bertatar	·tubartatar

PRETERITE PASSIVE		
pss	do·breth	*·tubrath*
3p	do·bretha	*·tubratha*

Figure B.6 – Paradigm for *do·beir* (S1a) (continued). Taken from Green (1995: 39), who adds that forms with * are attested for *beirid* or other compounds of *·beir*, e.g., *ar·beir*, *as·beir*, *con·beir*.



DO·BEIR: perfective forms based on *do·rat*, ‘give’

PRESENT INDICATIVE (A II)

1s	*do·rataim*	*·tartaim*
2s	*do·ratai*	*·tartai*
3s	*do·ratai*	·tartai
1p	*do·ratam*	*·tartam*
2p	*do·rataid*	*·tartaid*
3p	*do·ratat*	·tartat
pss	*do·ratather*	*·tartather*
3p	*do·ratater*	*·tartater*

IMPERFECT INDICATIVE

1s		*·tartainn*
2s		*·tartae*
3s		*·tartad*
1p		*·tartamais*
2p		*·tartathae*
3p		·tartatais
pss		*·tartathae*
3p		*·tartatais*

IMPERATIVE

2s	*d-a·rat*	*tart*
3s	*d-a·ratad*	*tartad*
1p	*d-a·ratam*	*tartam*
2p	*d-a·rataid*	*tartaid*
3p	*d-a·ratat*	*tartat*
pss	*d-a·ratather*	*tartather*
3p		*tartater*

PRESENT SUBJUNCTIVE (ā)

1s	*do·rat*	·tart
2s	*do·ratae*	*·tartae*
3s	do·rata	·tarta
1p	*do·ratam*	·tartam
2p	do·rataid	·tartaid
3p	do·ratat	·tartat
pss	do·ratar	·tartar
3p	do·rataiter	·tartaiter

PAST SUBJUNCTIVE

1s	*do·ratainn*	·tartainn
2s	*do·ratta*	*·tarta*
3s	*do·ratad*	·tartad
1p	*do·ratmais*	*·tartmais*
2p	*do·rattae*	*·tartae*
3p	*do·rattais*	*·tartais*
pss	*do·rattae*	·tartae
3p	*do·rattais*	*·tartais*

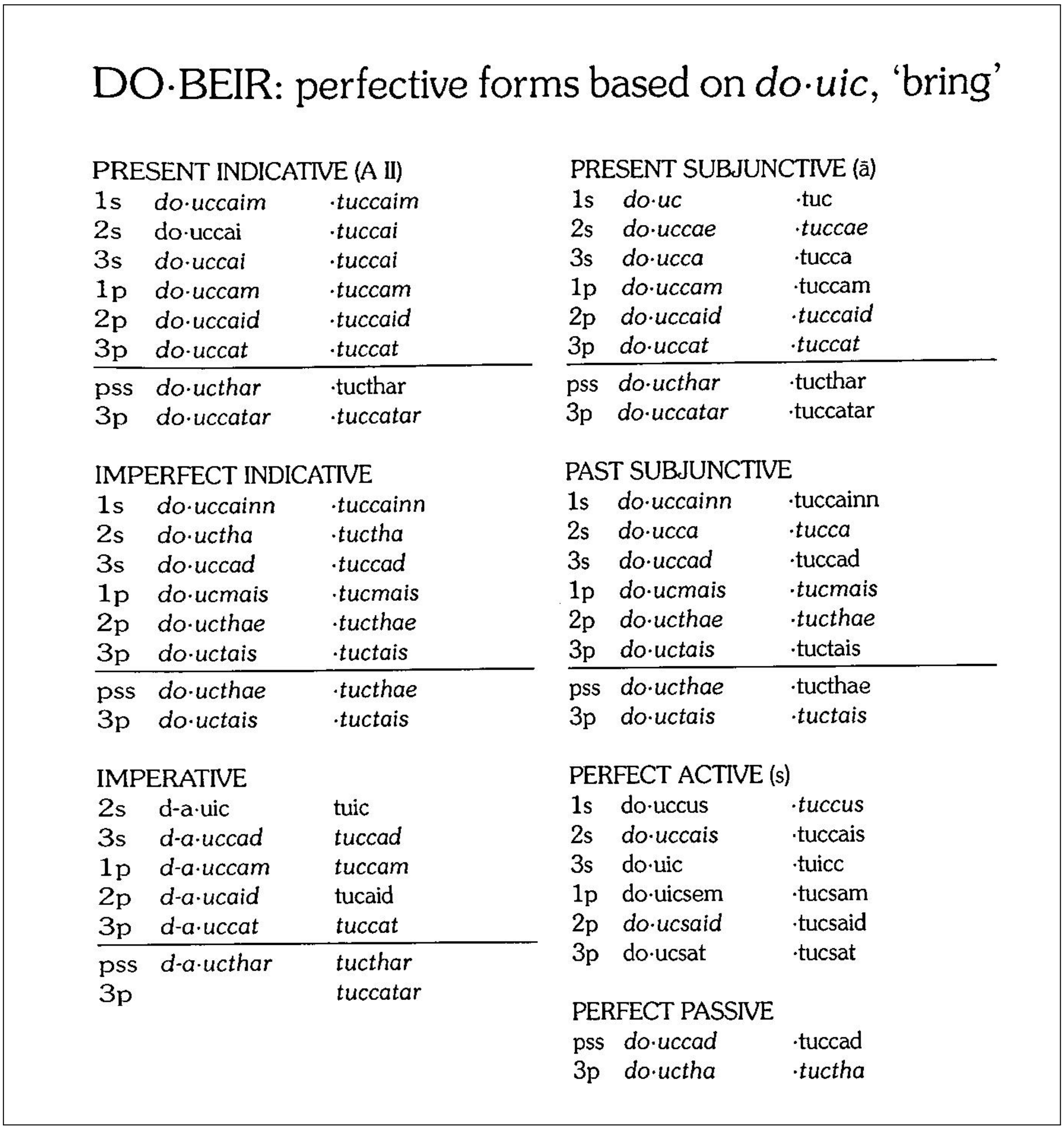
PERFECT ACTIVE (s)

1s	do·ratus	·tartus
2s	do·ratais	·tartais
3s	do·rat	·tarat
1p	do·ratsam	*·tartsam*
2p	do·ratsaid	·tartsaid
3p	do·ratsat	·tartsat, ·tartaisset

PERFECT PASSIVE

pss	do·ratad	·tartad
3p	do·ratta	·tarta

Figure B.7 – Paradigm for *do·ratai* (W2a, indep. perf. active 3sg. *do·rat*), suppletive stem of *do·beir*. Taken from Green (1995: 40).



DO·BEIR: perfective forms based on *do·uic*, ‘bring’

PRESENT INDICATIVE (A II)

1s	*do·uccaim*	*·tuccaim*
2s	do·uccai	*·tuccai*
3s	*do·uccai*	*·tuccai*
1p	*do·uccam*	*·tuccam*
2p	*do·uccaid*	*·tuccaid*
3p	*do·uccat*	*·tuccat*
pss	*do·ucthar*	·tucthar
3p	*do·uccatar*	*·tuccatar*

IMPERFECT INDICATIVE

1s	*do·uccainn*	*·tuccainn*
2s	*do·uctha*	*·tuctha*
3s	*do·uccad*	*·tuccad*
1p	*do·ucmais*	*·tucmais*
2p	*do·ucthae*	*·tucthae*
3p	*do·uctais*	*·tuctais*
pss	*do·ucthae*	*·tucthae*
3p	*do·uctais*	*·tuctais*

IMPERATIVE

2s	d-a·uic	tuic
3s	*d-a·uccad*	*tuccad*
1p	*d-a·uccam*	*tuccam*
2p	*d-a·ucaid*	tucaid
3p	*d-a·uccat*	*tuccat*
pss	*d-a·ucthar*	*tucthar*
3p		*tuccatar*

PRESENT SUBJUNCTIVE (ā)

1s	*do·uc*	·tuc
2s	*do·uccae*	*·tuccae*
3s	*do·ucca*	·tucca
1p	*do·uccam*	·tuccam
2p	*do·uccaid*	*·tuccaid*
3p	*do·uccat*	*·tuccat*
pss	*do·ucthar*	·tucthar
3p	*do·uccatar*	·tuccatar

PAST SUBJUNCTIVE

1s	*do·uccainn*	·tuccainn
2s	*do·ucca*	*·tucca*
3s	*do·uccad*	·tuccad
1p	*do·ucmais*	*·tucmais*
2p	*do·ucthae*	*·tucthae*
3p	*do·uctais*	·tuctais
pss	*do·ucthae*	·tucthae
3p	*do·uctais*	*·tuctais*

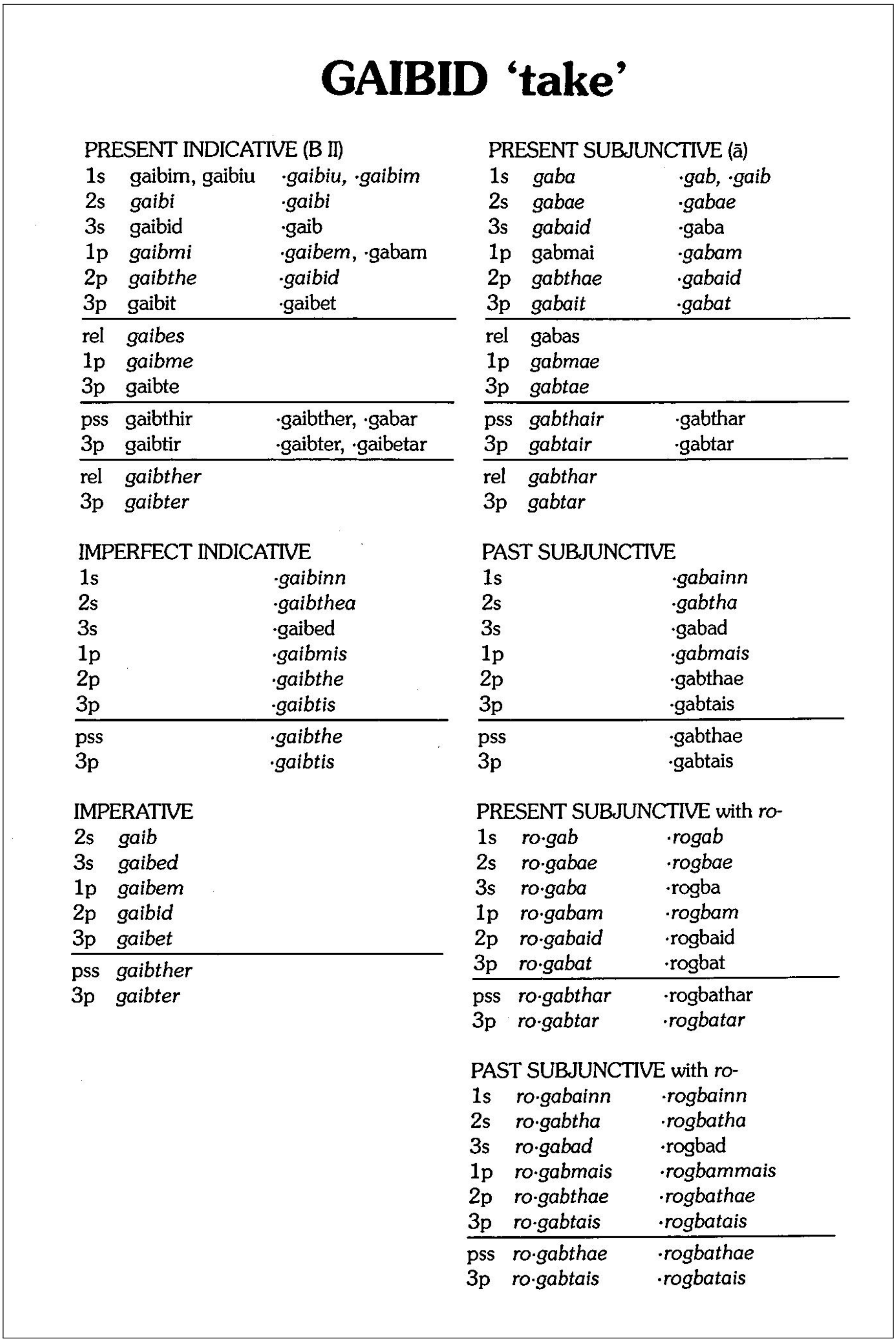
PERFECT ACTIVE (s)

1s	do·uccus	*·tuccus*
2s	*do·uccais*	·tuccais
3s	do·uic	·tuicc
1p	do·uicsem	·tucsam
2p	*do·ucsaid*	·tucsaid
3p	do·ucsat	·tucsat

PERFECT PASSIVE

pss	*do·uccad*	·tuccad
3p	*do·uctha*	*·tuctha*

Figure B.8 – Paradigm for *do·uccai* (W2a, indep. perf. active 3sg. *do·uic*), suppletive stem of *do·beir*. Taken from Green (1995: 41).



GAIBID ‘take’

PRESENT INDICATIVE (B II)

1s	gaibim, gaibiu	*·gaibiu, ·gaibim*
2s	*gaibi*	*·gaibi*
3s	gaibid	·gaib
1p	*gaibmi*	*·gaibem*, ·gabam
2p	*gaibthe*	*·gaibid*
3p	gaibit	·gaibet
rel	*gaibes*	
1p	*gaibme*	
3p	gaibte	
pss	gaibthir	·gaibther, ·gabar
3p	gaibtir	·gaibter, ·gaibetar
rel	*gaibther*	
3p	*gaibter*	

PRESENT SUBJUNCTIVE (ā)

1s	*gaba*	*·gab, ·gaib*
2s	*gabae*	*·gabae*
3s	*gabaid*	·gaba
1p	gabmai	*·gabam*
2p	gabthae	*·gabaid*
3p	*gabait*	*·gabat*
rel	gabas	
1p	*gabmae*	
3p	*gabtae*	
pss	*gabthair*	·gabthar
3p	*gabtair*	·gabtar
rel	*gabthar*	
3p	*gabtar*	

IMPERFECT INDICATIVE

1s	*·gaibinn*
2s	*·gaibthea*
3s	·gaibed
1p	*·gaibmis*
2p	*·gaibthe*
3p	*·gaibtis*
pss	*·gaibthe*
3p	*·gaibtis*

PAST SUBJUNCTIVE

1s	*·gabainn*
2s	*·gabtha*
3s	·gabad
1p	*·gabmais*
2p	·gabthae
3p	·gabtais
pss	·gabthae
3p	·gabtais

IMPERATIVE

2s	*gaib*
3s	*gaibed*
1p	*gaibem*
2p	*gaibid*
3p	*gaibet*
pss	*gaibther*
3p	*gaibter*

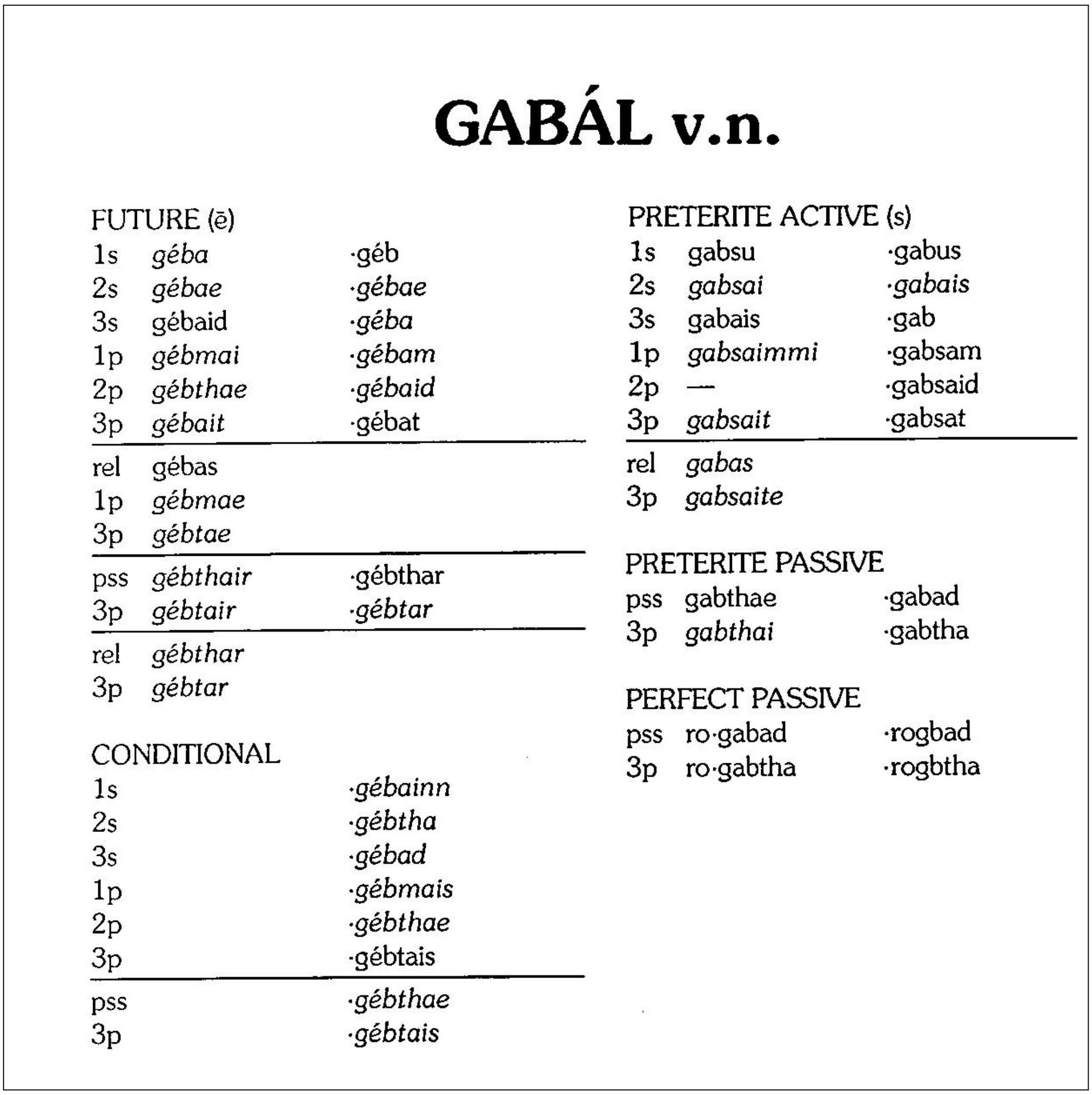
PRESENT SUBJUNCTIVE with *ro-*

1s	*ro·gab*	*·rogab*
2s	*ro·gabae*	*·rogbae*
3s	*ro·gaba*	·rogba
1p	*ro·gabam*	*·rogbam*
2p	*ro·gabaid*	·rogbaid
3p	*ro·gabat*	·rogbat
pss	*ro·gabthar*	·rogbathar
3p	*ro·gabtar*	*·rogbatar*

PAST SUBJUNCTIVE with *ro-*

1s	*ro·gabainn*	*·rogbainn*
2s	*ro·gabtha*	*·rogbatha*
3s	*ro·gabad*	·rogbad
1p	*ro·gabmais*	*·rogbammais*
2p	*ro·gabthae*	*·rogbathae*
3p	*ro·gabtais*	*·rogbatais*
pss	*ro·gabthae*	*·rogbathae*
3p	*ro·gabtais*	*·rogbatais*

Figure B.9 – Paradigm for *gaibid* (S2). Taken from Green (1995: 60).



GABÁL v.n.

FUTURE (ē)

1s	*géba*	·géb
2s	*gébae*	*·gébae*
3s	gébaid	*·géba*
1p	*gébmai*	*·gébam*
2p	*gébthae*	*·gébaid*
3p	*gébait*	·gébat
rel	gébas	
1p	*gébmae*	
3p	*gébtae*	
pss	*gébthair*	·gébthar
3p	*gébtair*	*·gébtar*
rel	*gébthar*	
3p	*gébtar*	

CONDITIONAL

1s		*·gébainn*
2s		*·gébtha*
3s		*·gébad*
1p		*·gébmais*
2p		*·gébthae*
3p		·gébtais
pss		*·gébthae*
3p		*·gébtais*

PRETERITE ACTIVE (s)

1s	gabsu	·gabus
2s	*gabsai*	*·gabais*
3s	gabais	·gab
1p	*gabsaimmi*	·gabsam
2p	—	·gabsaid
3p	*gabsait*	·gabsat
rel	*gabas*	
3p	*gabsaite*	

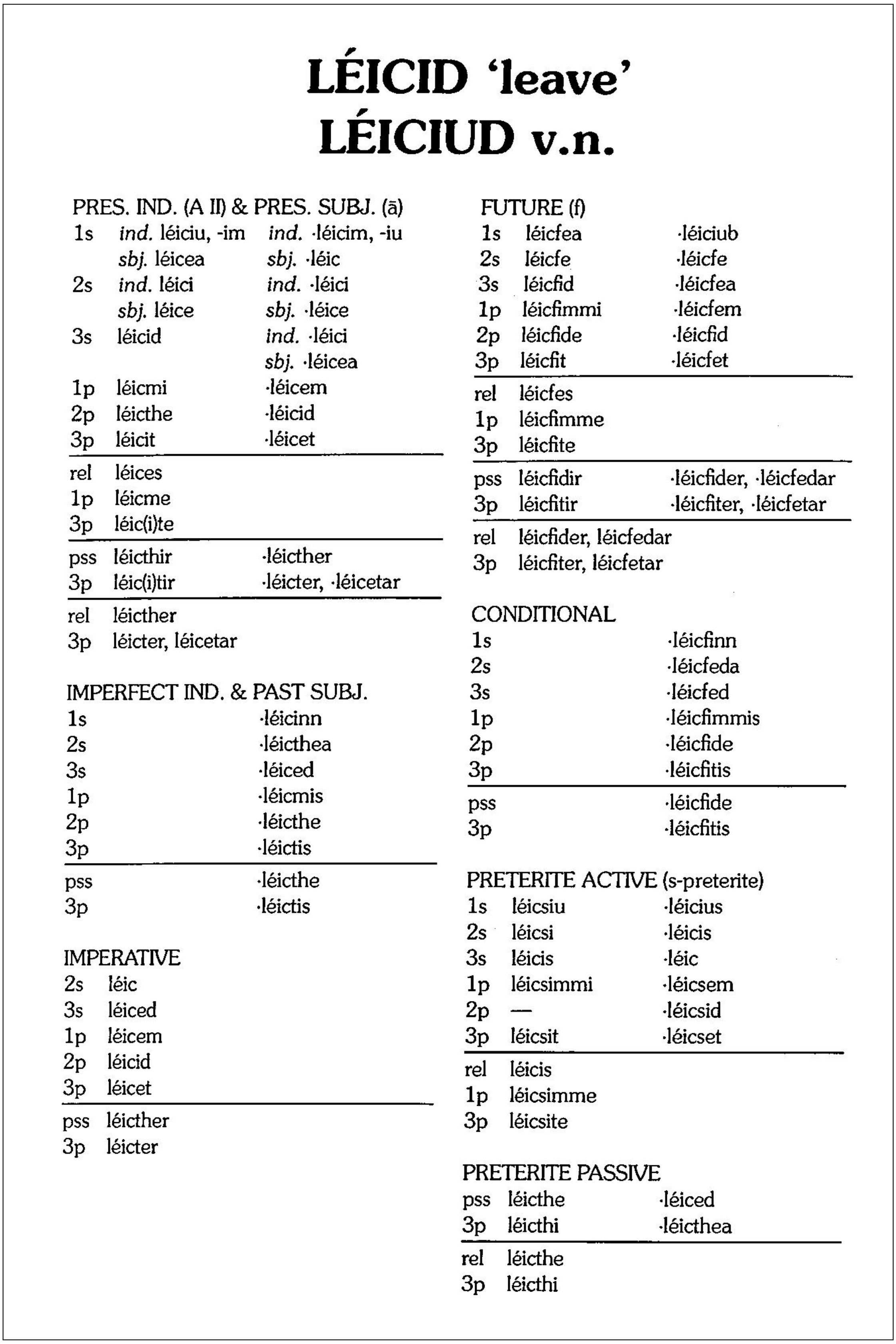
PRETERITE PASSIVE

pss	gabthae	·gabad
3p	*gabthai*	·gabtha

PERFECT PASSIVE

pss	ro·gabad	·rogbad
3p	ro·gabtha	·rogbtha

Figure B.10 – Paradigm for *gaibid* (S2) (continued). Taken from Green (1995: 61).



LÉICID 'leave'
LÉICIUD v.n.

PRES. IND. (A II) & PRES. SUBJ. (ā)

1s	*ind.* léiciu, -im	*ind.* ·léicim, -iu
	sbj. léicea	*sbj.* ·léic
2s	*ind.* léici	*ind.* ·léici
	sbj. léice	*sbj.* ·léice
3s	léicid	*ind.* ·léici
		sbj. ·léicea
1p	léicmi	·léicem
2p	léicthe	·léicid
3p	léicit	·léicet
rel	léices	
1p	léicme	
3p	léic(i)te	
pss	léicthir	·léicther
3p	léic(i)tir	·léicter, ·léicetar
rel	léicther	
3p	léicter, léicetar	

IMPERFECT IND. & PAST SUBJ.

1s	·léicinn
2s	·léicthea
3s	·léiced
1p	·léicmis
2p	·léicthe
3p	·léictis
pss	·léicthe
3p	·léictis

IMPERATIVE

2s	léic
3s	léiced
1p	léicem
2p	léicid
3p	léicet
pss	léicther
3p	léicter

FUTURE (f)

1s	léicfea	·léiciub
2s	léicfe	·léicfe
3s	léicfid	·léicfea
1p	léicfimmi	·léicfem
2p	léicfide	·léicfid
3p	léicfit	·léicfet
rel	léicfes	
1p	léicfimme	
3p	léicfite	
pss	léicfidir	·léicfider, ·léicfedar
3p	léicfitir	·léicfiter, ·léicfetar
rel	léicfider, léicfedar	
3p	léicfiter, léicfetar	

CONDITIONAL

1s	·léicfinn
2s	·léicfeda
3s	·léicfed
1p	·léicfimmis
2p	·léicfide
3p	·léicfitis
pss	·léicfide
3p	·léicfitis

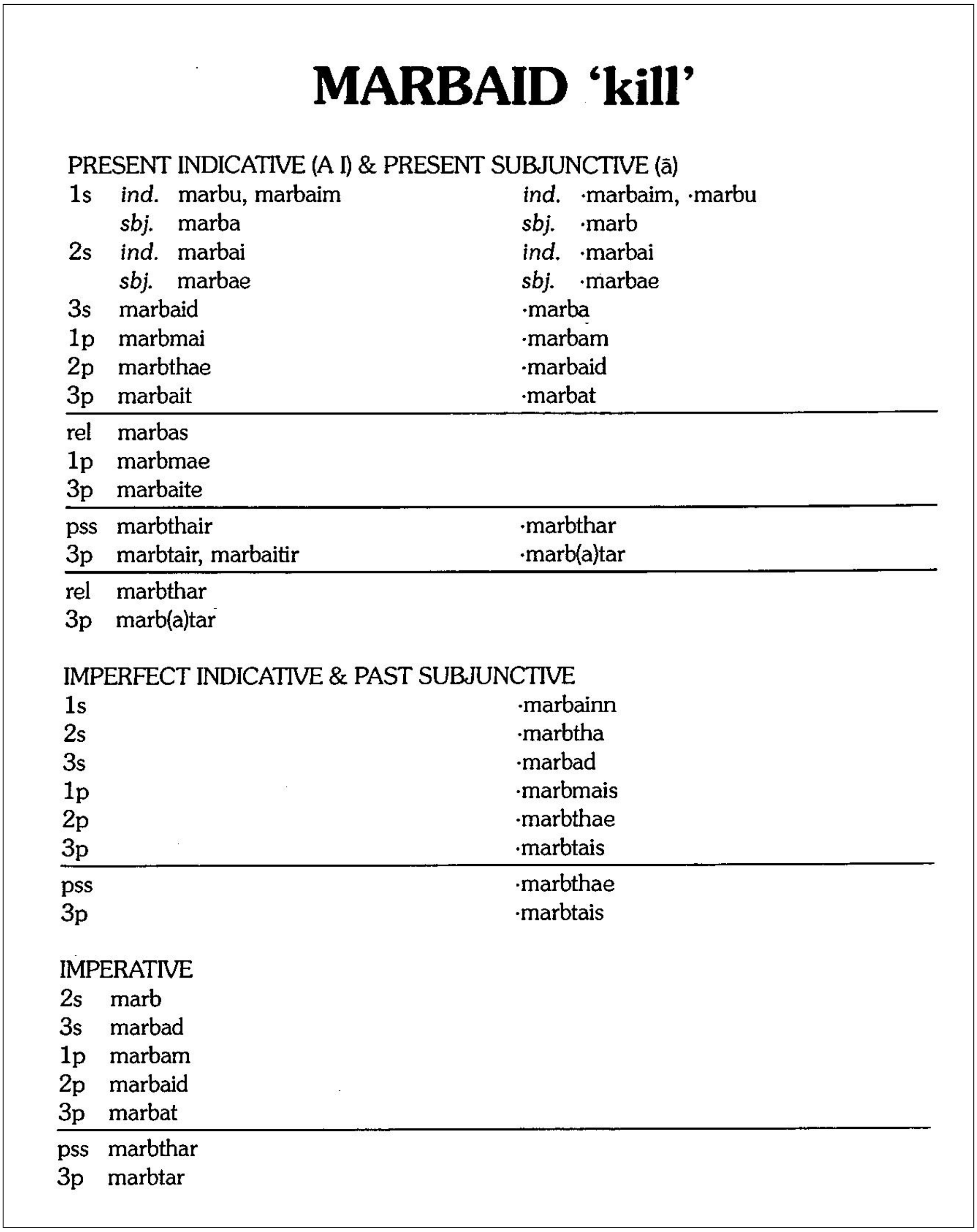
PRETERITE ACTIVE (s-preterite)

1s	léicsiu	·léicius
2s	léicsi	·léicis
3s	léicis	·léic
1p	léicsimmi	·léicsem
2p	—	·léicsid
3p	léicsit	·léicset
rel	léicis	
1p	léicsimme	
3p	léicsite	

PRETERITE PASSIVE

pss	léicthe	·léiced
3p	léicthi	·léicthea
rel	léicthe	
3p	léicthi	

Figure B.11 – Paradigm for *léicid* (W2a). Taken from Green (1995: 73).



MARBAID 'kill'

PRESENT INDICATIVE (A I) & PRESENT SUBJUNCTIVE (ā)

1s	*ind.*	marbu, marbaim	*ind.*	·marbaim, ·marbu
	sbj.	marba	*sbj.*	·marb
2s	*ind.*	marbai	*ind.*	·marbai
	sbj.	marbae	*sbj.*	·marbae
3s		marbaid		·marba
1p		marbmai		·marbam
2p		marbthae		·marbaid
3p		marbait		·marbat
rel		marbas		
1p		marbmae		
3p		marbaite		
pss		marbthair		·marbthar
3p		marbtair, marbaitir		·marb(a)tar
rel		marbthar		
3p		marb(a)tar		

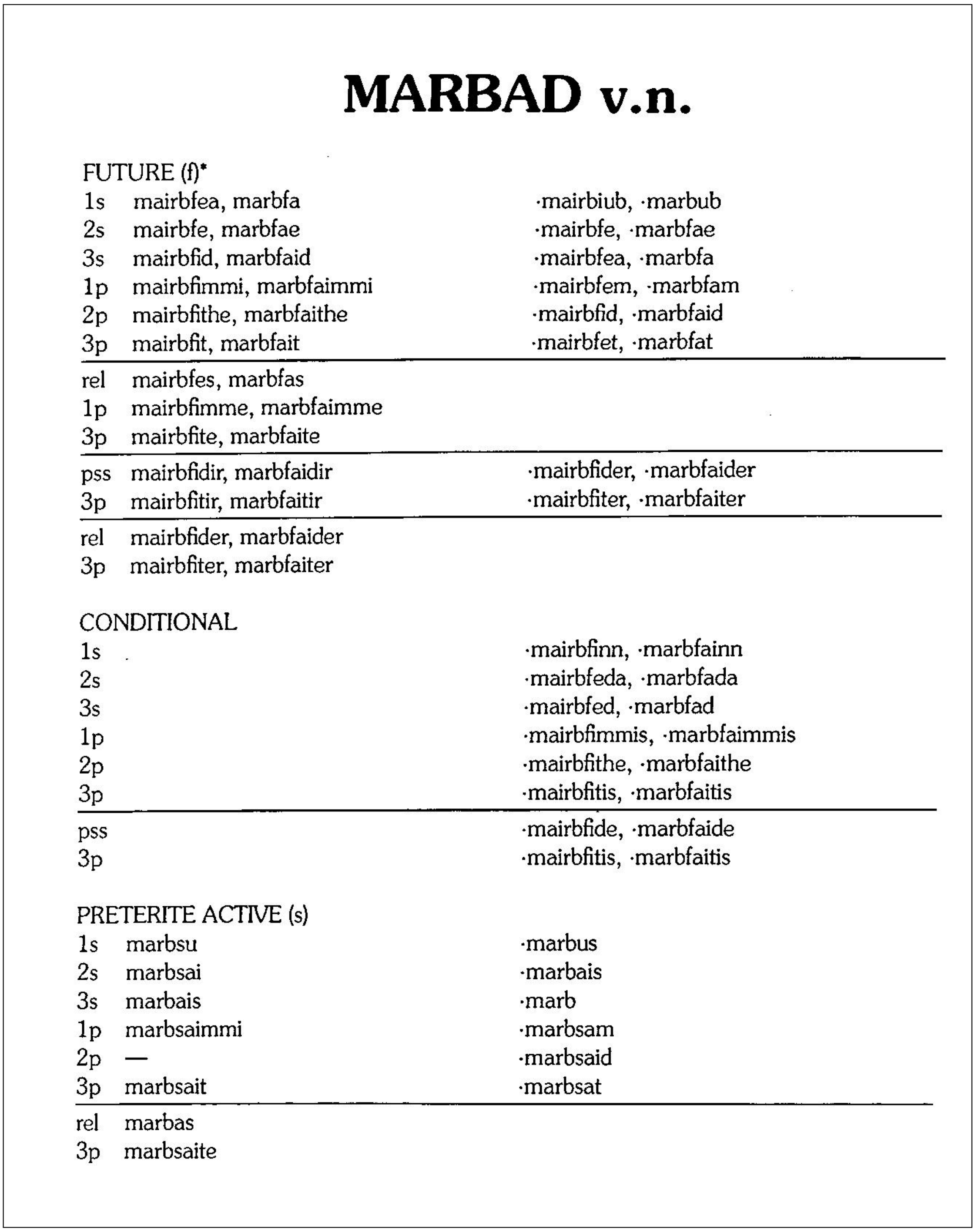
IMPERFECT INDICATIVE & PAST SUBJUNCTIVE

1s	·marbainn
2s	·marbtha
3s	·marbad
1p	·marbmais
2p	·marbthae
3p	·marbtais
pss	·marbthae
3p	·marbtais

IMPERATIVE

2s	marb
3s	marbad
1p	marbam
2p	marbaid
3p	marbat
pss	marbthar
3p	marbtar

Figure B.12 – Paradigm for *marbaid* (W1). Taken from Green (1995: 76).



MARBAD v.n.

FUTURE (f)*

1s	mairbfea, marbfa	·mairbiub, ·marbub
2s	mairbfe, marbfae	·mairbfe, ·marbfae
3s	mairbfid, marbfaid	·mairbfea, ·marbfa
1p	mairbfimmi, marbfaimmi	·mairbfem, ·marbfam
2p	mairbfithe, marbfaithe	·mairbfid, ·marbfaid
3p	mairbfit, marbfait	·mairbfet, ·marbfat
rel	mairbfes, marbfas	
1p	mairbfimme, marbfaimme	
3p	mairbfite, marbfaite	
pss	mairbfidir, marbfaidir	·mairbfider, ·marbfaider
3p	mairbfitir, marbfaitir	·mairbfiter, ·marbfaiter
rel	mairbfider, marbfaider	
3p	mairbfiter, marbfaiter	

CONDITIONAL

1s		·mairbfinn, ·marbfainn
2s		·mairbfeda, ·marbfada
3s		·mairbfed, ·marbfad
1p		·mairbfimmis, ·marbfaimmis
2p		·mairbfithe, ·marbfaithe
3p		·mairbfitis, ·marbfaitis
pss		·mairbfide, ·marbfaide
3p		·mairbfitis, ·marbfaitis

PRETERITE ACTIVE (s)

1s	marbsu	·marbus
2s	marbsai	·marbais
3s	marbais	·marb
1p	marbsaimmi	·marbsam
2p	—	·marbsaid
3p	marbsait	·marbsat
rel	marbas	
3p	marbsaite	

Figure B.13 – Paradigm for *marbaid* (W1) (continued). Taken from Green (1995: 77), who adds the following note for *: ‘In the *f*-future and conditional of class AI verbs, the consonant preceding the *f* is usually palatalized, but neutral quality is also often found. Both variations are shown here. See GOI § 636’ (GOI = Thurneysen 1946).

Appendix C

Code

The code is also available online.[1]

C.1 Lexicons (`.lexc`)

C.1.1 `copula.lexc`

```
!***** copula.lexc *****
! Th. Fransen, 18/08/19

!\\\\\ DECLARE MULTICHAR SYMBOLS /////

Multichar_Symbols

! *UPPER symbols (Tags)*

+
+1P
+2P
+3P
+ABS
+AUG
+COND
+CONJ
+CONJ_PART
+CONSUETUD
+COP
+DEPEND
+FUT
+IPF
+IMP
+IND
```

[1] https://github.com/ThFransen84.

```
+LEN
+NEG
+PAST
+PL
+PRS
+REL
+SG
+SUBJ
+VROOT

!\\\\\ BEGIN CONTINUATION CLASSES /////

!*** Root = start ***

LEXICON Root
    Independent;
    Prefix;

LEXICON Independent
    Present;
    Imperative;
    presSubj;
    pastSubj;
    Future;
    Conditional;
    Past;

LEXICON Prefix
                            Dependent;
ní+CONJ_PART+NEG:ní         Dependent;
                            depImp;
ná+CONJ_PART+IMP+NEG:ná     depImp;

LEXICON Dependent
    depPres;
    depCons;
    depPresSubj;
    depPastSubj;
    depFut;
    depCond;
    depPast;

LEXICON Present
is+VROOT+COP+PRS+IND+1P+SG:am           #;
is+VROOT+COP+PRS+IND+2P+SG:at           #;
is+VROOT+COP+PRS+IND+2P+SG:it           #;
is+VROOT+COP+PRS+IND+3P+SG:is           #;
is+VROOT+COP+PRS+IND+3P+SG+REL:as       #;
```

```
is+VROOT+COP+PRS+IND+1P+PL:ammi         #;
is+VROOT+COP+PRS+IND+1P+PL:ammin        #;
is+VROOT+COP+PRS+IND+2P+PL:adi          #;
is+VROOT+COP+PRS+IND+2P+PL:adib         #;
is+VROOT+COP+PRS+IND+3P+PL:it           #;
is+VROOT+COP+PRS+IND+3P+PL+REL:ata      #;
is+VROOT+COP+PRS+IND+3P+PL+REL:at       #;

LEXICON Imperative
bí+VROOT+COP+IMP+2P+SG+LEN:ba       #;
bí+VROOT+COP+IMP+3P+SG+LEN:bed      #;
bí+VROOT+COP+IMP+3P+SG+LEN:bad      #;
bí+VROOT+COP+IMP+1P+PL+LEN:baan     #;
bí+VROOT+COP+IMP+1P+PL+LEN:ban      #;
bí+VROOT+COP+IMP+2P+PL+LEN:bed      #;
bí+VROOT+COP+IMP+2P+PL+LEN:bad      #;
bí+VROOT+COP+IMP+3P+PL+LEN:bat      #;

LEXICON presSubj
bí+VROOT+COP+PRS+SUBJ+1P+SG:ba              #;
bí+VROOT+COP+PRS+SUBJ+2P+SG:ba              #;
bí+VROOT+COP+PRS+SUBJ+2P+SG:be              #;
bí+VROOT+COP+PRS+SUBJ+3P+SG:ba              #;
bí+VROOT+COP+PRS+SUBJ+3P+SG+REL:bes         #;
bí+VROOT+COP+PRS+SUBJ+3P+SG+REL:bas         #;
bí+VROOT+COP+PRS+SUBJ+2P+PL:bede            #;
bí+VROOT+COP+PRS+SUBJ+3P+PL+REL+LEN:bete    #;
bí+VROOT+COP+PRS+SUBJ+3P+PL+REL+LEN:beta    #;

LEXICON pastSubj
bí+VROOT+COP+PAST+SUBJ+3P+SG+LEN:bed            #;
bí+VROOT+COP+PAST+SUBJ+3P+SG+LEN:bad            #;
bí+VROOT+COP+PAST+SUBJ+3P+SG+LEN:bid            #;
bí+VROOT+COP+PAST+SUBJ+1P+PL:bemmis             #;
bí+VROOT+COP+PAST+SUBJ+3P+PL:betis              #;
bí+VROOT+COP+PAST+SUBJ+3P+PL:bitis              #;
ro+AUG+DEPEND+bí+VROOT+COP+COND+3P+PL:roptis    #;

LEXICON Future
bí+VROOT+COP+FUT+1P+SG:be               #;
bí+VROOT+COP+FUT+2P+SG:be               #;
bí+VROOT+COP+FUT+3P+SG:bid              #;
bí+VROOT+COP+FUT+3P+SG+REL+LEN:bes      #;
bí+VROOT+COP+FUT+3P+SG+REL+LEN:bas      #;
bí+VROOT+COP+FUT+1P+PL:bimmi            #;
bí+VROOT+COP+FUT+1P+PL:bemmi            #;
bí+VROOT+COP+FUT+3P+PL:bit              #;
bí+VROOT+COP+FUT+3P+PL+REL+LEN:beta     #;
```

```
bí+VROOT+COP+FUT+3P+PL+REL+LEN:bat      #;

LEXICON Conditional
bí+VROOT+COP+COND+3P+SG+LEN:bed     #;
bí+VROOT+COP+COND+3P+PL:beitis      #;

LEXICON Past
bí+VROOT+COP+PAST+1P+SG:basa    #;
bí+VROOT+COP+PAST+3P+SG:ba      #;
bí+VROOT+COP+PAST+3P+PL:batir   #;
bí+VROOT+COP+PAST+3P+PL:batar   #;

ro+AUG+DEPEND+bí+VROOT+COP+PAST+1P+SG:ro-bsa            #;
ro+AUG+DEPEND+bí+VROOT+COP+PAST+2P+SG:ro-psa            #;
ro+AUG+DEPEND+bí+VROOT+COP+PRS+IND+3P+SG+LEN:ro-po      #;
ro+AUG+DEPEND+bí+VROOT+COP+PRS+IND+3P+SG+LEN:ro-bo      #;
ro+AUG+DEPEND+bí+VROOT+COP+PRS+IND+3P+SG+LEN:ro-pu      #;
ro+AUG+DEPEND+bí+VROOT+COP+PRS+IND+3P+SG+LEN:ro-bu      #;
ro+AUG+DEPEND+bí+VROOT+COP+PAST+3P+PL:ro-bummar         #;
ro+AUG+DEPEND+bí+VROOT+COP+PAST+3P+PL:ro-ptar           #;

LEXICON depPres
+DEPEND+tá+VROOT+COP+PRS+IND+1P+SG+LEN:-ta      #;
+DEPEND+tá+VROOT+COP+PRS+IND+1P+SG+LEN:-da      #;
+DEPEND+tá+VROOT+COP+PRS+IND+2P+SG+LEN:-ta      #;
+DEPEND+tá+VROOT+COP+PRS+IND+2P+SG+LEN:-da      #;
+DEPEND+tá+VROOT+COP+PRS+IND+1P+PL+LEN:-tan     #;
+DEPEND+tá+VROOT+COP+PRS+IND+1P+PL+LEN:-dan     #;
+DEPEND+tá+VROOT+COP+PRS+IND+2P+PL+LEN:-tad     #;
+DEPEND+tá+VROOT+COP+PRS+IND+2P+PL+LEN:-dad     #;
+DEPEND+tá+VROOT+COP+PRS+IND+3P+PL+LEN:-tat     #;
+DEPEND+tá+VROOT+COP+PRS+IND+3P+PL+LEN:-dat     #;

LEXICON depImp
+DEPEND+bí+VROOT+COP+IMP+3P+SG+LEN:-bad     #;
+DEPEND+bí+VROOT+COP+IMP+2P+PL+LEN:-bad     #;
+DEPEND+bí+VROOT+COP+IMP+3P+PL+LEN:-bat     #;

LEXICON depCons
+DEPEND+bí+VROOT+COP+PRS+IND+CONSUETUD+3P+SG:-bi    #;
+DEPEND+bí+VROOT+COP+PRS+IND+CONSUETUD+3P+SG:-pi    #;

LEXICON depPresSubj
+DEPEND+bí+VROOT+COP+PRS+SUBJ+1P+SG+LEN:-ba     #;
+DEPEND+bí+VROOT+COP+PRS+SUBJ+2P+SG:-ba         #;
+DEPEND+bí+VROOT+COP+PRS+SUBJ+1P+PL:-ban        #;
+DEPEND+bí+VROOT+COP+PRS+SUBJ+2P+PL:-bad        #;
+DEPEND+bí+VROOT+COP+PRS+SUBJ+3P+PL+LEN:-pat    #;
```

```
+DEPEND+bí+VROOT+COP+PRS+SUBJ+3P+PL+LEN:-bat    #;

LEXICON depPastSubj
+DEPEND+bí+VROOT+COP+PAST+SUBJ+1P+SG:-benn      #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+1P+SG:-bin       #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+2P+SG:-ptha      #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+3P+SG+LEN:-bed   #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+3P+SG+LEN:-bad   #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+1P+PL:-bimmis    #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+3P+PL:-btis      #;
+DEPEND+bí+VROOT+COP+PAST+SUBJ+3P+PL:-ptis      #;

LEXICON depFut
+DEPEND+bí+VROOT+COP+FUT+3P+SG:-ba      #;
+DEPEND+bí+VROOT+COP+FUT+3P+SG:-pa      #;
+DEPEND+bí+VROOT+COP+FUT+3P+PL:-bat     #;

LEXICON depCond
+DEPEND+bí+VROOT+COP+COND+3P+SG+LEN:-bad    #;

LEXICON depPast
+DEPEND+bí+VROOT+COP+PAST+1P+SG:-psa        #;
+DEPEND+bí+VROOT+COP+PAST+3P+SG+LEN:-bu     #;
+DEPEND+bí+VROOT+COP+PAST+3P+SG+LEN:-pu     #;
+DEPEND+bí+VROOT+COP+PAST+3P+SG+LEN:-bo     #;
+DEPEND+bí+VROOT+COP+PAST+3P+SG+LEN:-po     #;
+DEPEND+bí+VROOT+COP+PAST+3P+PL:-btar       #;

+DEPEND+ro+AUG+bí+VROOT+COP+PAST+1P+SG:-rbsa        #;
+DEPEND+ro+AUG+bí+VROOT+COP+PAST+3P+SG+LEN:-rbo     #;
+DEPEND+ro+AUG+bí+VROOT+COP+PAST+3P+SG+LEN:-rbu     #;
+DEPEND+ro+AUG+bí+VROOT+COP+PAST+1P+PL:-rbommar     #;
+DEPEND+ro+AUG+bí+VROOT+COP+PAST+3P+PL:-rbtar       #;
```

C.1.2 `proclitic.lexc`

```
!***** proclitic.lexc *****
! Th. Fransen, 13/08/19

!\\\\\ DECLARE MULTICHAR SYMBOLS /////

Multichar_Symbols

! *UPPER symbols (Tags)*

+
+A
```

```
+B
+C
+1P
+2P
+3P
+AUG
+FEM
+H
+IMP
+INTERR
+LEN
+MASC
+NAS
+NEG
+NEUT
+CONJ_PART
+PL
+PRON
+PV1
+REL
+SG

! *LOWER symbols (Triggers)*

^M
^N
^PRONa

! *Flags*

@P.PART.NO@
@P.PV.AD@
@P.PV.ARE@
@P.PV.COM@
@P.PV.FO@
@P.PV.IMBI@
@P.PV.TO@
@P.PV.SV@   ! substantive verb

!\\\\\ BEGIN CONTINUATION CLASSES /////

!*** Root = start ***

LEXICON Root
                Preverb;
                conjPart;
@P.PART.NO@     No;
                Ro;
```

```

!*** Preverbs ***

LEXICON Preverb
@P.PV.AD@           AD;
@P.PV.ARE@          ARE;
@P.PV.COM@          COM;
@P.PV.FO@           FO;
@P.PV.IMBI@         IMBI;
@P.PV.TO@           TO;
@P.PV.SV@ad+PV1:at  #;  ! substantive verb

LEXICON AD
ad+PV1:ad           #;
ad+PV1:at           pronB;
ad+PV1+REL+LEN:ad   #;
ad+PV1+REL+NAS:ad   #;
ad+PV1+REL+LEN:a    pronC;
ad+PV1+REL+NAS:an   pronC;

LEXICON ARE
are+PV1:ar              #;
are+PV1:aru             pronA;
are+PV1:aro             pronA;
are+PV1:ari             pronA;
are+PV1:ara             pronA;
are+PV1+REL+LEN:ara     #;
are+PV1+REL+NAS:ara     #;
are+PV1+REL+LEN:ar      pronC;
are+PV1+REL+NAS:aran    pronC;

LEXICON COM
com+PV1:con             #;
com+PV1:cot             pronB;
com+PV1+REL+LEN:con     #;
com+PV1+REL+NAS:con     #;
com+PV1+REL+LEN:co      pronC;
com+PV1+REL+NAS:con     pronC;

LEXICON FO
fo+PV1:fo           #;
fo+PV1:fo           pronA;
fo+PV1+REL+LEN:fo   #;
fo+PV1+REL+NAS:fo   #;
fo+PV1+REL+LEN:fo   pronC;
fo+PV1+REL+NAS:fon  pronC;

LEXICON IMBI
```

```
imbi+PV1:i^M                #;
imbi+PV1:i^Mu               pronA;
imbi+PV1:i^Mi               pronA;
imbi+PV1+REL+LEN:i^Me       #;
imbi+PV1+REL+NAS:i^Me       #;
imbi+PV1+REL+LEN:i^Mu       pronC;
imbi+PV1+REL+LEN:i^Mi       pronC;
imbi+PV1+REL+NAS:i^Mun      pronC;
imbi+PV1+REL+NAS:i^Min      pronC;

LEXICON TO
to+PV1:do           #;
to+PV1:do           pronA;
to+PV1+REL+LEN:do   #;
to+PV1+REL+NAS:do   #;
to+PV1+REL+LEN:do   pronC;
to+PV1+REL+NAS:don  pronC;

!*** Conjunct particles ***

LEXICON conjPart
ní+CONJ_PART+NEG:ní     #;
ní+CONJ_PART+NEG:ní     pronA;
ní+CONJ_PART+NEG:ní     roNonRel;

ná+CONJ_PART+IMP+NEG:ná                             #;
ná+CONJ_PART+IMP+NEG:nach                           pronNach;
ná+CONJ_PART+IMP+NEG+PRON+B+3P+SG+NEUT+LEN:nadid    #;

in+CONJ_PART+INTERR+NAS:in  #;
in+CONJ_PART+INTERR:in      pronC;
in+CONJ_PART+INTERR:in      roNonRel;

nád+CONJ_PART+INTERR+NEG:nád                            #;
nád+CONJ_PART+INTERR+NEG:ná                             roNonRel;
nád+CONJ_PART+INTERR+NEG+PRON+C+3P+SG+NEUT+LEN:nadid    #;
nád+CONJ_PART+INTERR+NEG:innach                         pronNach;

nád+CONJ_PART+REL+NEG+LEN:nád                       #;
nád+CONJ_PART+REL+NEG+NAS:nád                       #;
nád+CONJ_PART+REL+NEG:ná                            roRel;
nád+CONJ_PART+REL+NEG:nach                          pronNach;
nád+CONJ_PART+REL+NEG+PRON+C+3P+SG+NEUT+LEN:nadid   #;

!*** No ***

LEXICON No
no+CONJ_PART:no             #;
```

```
no+CONJ_PART+REL+LEN:no     #;
no+CONJ_PART+REL+NAS:no     #;
no+CONJ_PART:no             pronA;
no+CONJ_PART+REL+LEN:no     pronC;
no+CONJ_PART+REL+NAS:non    pronC;

!*** Ro ***

LEXICON Ro
    roNonRel;
    roRel;

LEXICON roNonRel
+ro+AUG:ro  #;
+ro+AUG:ro  pronA;

LEXICON roRel
+ro+AUG+REL+LEN:ro      #;
+ro+AUG+REL+NAS:ro      #;
+ro+AUG+REL+LEN:ro      pronC;
+ro+AUG+REL+NAS:ron     pronC;

!\\\\\ INFIXED PRONOUNS /////

LEXICON pronA
+PRON+A+1P+SG+LEN:^M            #;
+PRON+A+2P+SG+LEN:t             #;
+PRON+A+3P+SG+MASC+NAS:^PRONa   #;
+PRON+A+3P+SG+FEM:s             #;
+PRON+A+3P+SG+FEM+NAS:s         #;
+PRON+A+3P+SG+NEUT+LEN:^PRONa   #;
+PRON+A+1P+PL:^N                #;
+PRON+A+2P+PL:b                 #;
+PRON+A+3P+PL:s                 #;
+PRON+A+3P+PL+NAS:s             #;

LEXICON pronB
+PRON+B+1P+SG+LEN:om        #;
+PRON+B+1P+SG+LEN:um        #;
+PRON+B+1P+SG+LEN:am        #;
+PRON+B+1P+SG+LEN:amm       #;
+PRON+B+2P+SG+LEN:ot        #;
+PRON+B+2P+SG+LEN:at        #;
+PRON+B+2P+SG+LEN:0         #;
+PRON+B+3P+SG+MASC+NAS:0    #;
+PRON+B+3P+SG+MASC+NAS:a    #;
+PRON+B+3P+SG+FEM+H:a       #;
+PRON+B+3P+SG+NEUT+LEN:0    #;
```

```
+PRON+B+1P+PL:an            #;
+PRON+B+1P+PL:ann           #;
+PRON+B+2P+PL:ob            #;
+PRON+B+2P+PL:ab            #;
+PRON+B+3P+PL+H:a           #;

LEXICON pronC
+PRON+C+1P+SG+LEN:dom       #;
+PRON+C+1P+SG+LEN:dum       #;
+PRON+C+1P+SG+LEN:dam       #;
+PRON+C+1P+SG+LEN:damm      #;
+PRON+C+2P+SG+LEN:dat       #;
+PRON+C+2P+SG+LEN:dit       #;
+PRON+C+3P+SG+FEM+H:da      #;
+PRON+C+1P+PL:don           #;
+PRON+C+1P+PL:dun           #;
+PRON+C+1P+PL:din           #;
+PRON+C+1P+PL:dan           #;
+PRON+C+1P+PL:dann          #;
+PRON+C+2P+PL:dob           #;
+PRON+C+2P+PL:dub           #;
+PRON+C+2P+PL:dib           #;
+PRON+C+2P+PL:dab           #;
+PRON+C+3P+PL+H:da          #;

LEXICON pronNach
+PRON+1P+SG+LEN:am          #;
+PRON+1P+SG+LEN:im          #;
+PRON+2P+SG+LEN:at          #;
+PRON+2P+SG+LEN:it          #;
+PRON+3P+SG+MASC+NAS:       #;
+PRON+3P+SG+FEM+H:a         #;
+PRON+3P+SG+NEUT+LEN:       #;
+PRON+3P+SG+NEUT+LEN:id     #;
+PRON+1P+PL:an              #;
+PRON+2P+PL:ab              #;
+PRON+3P+PL+H:a             #;
```

C.1.3 se.lexc

Technically, this file is se_empty.lexc, with placeholders for specific stem lists. Stem entries are maintained in separate files (section C.4, p. 222). Cf. section C.5 (p. 224) for a shell script that inserts stems into se_empty.lexc and creates a file se.lexc.

```
!***** se.lexc *****
! Th. Fransen, 13/08/19

```

```
!\\\\\ DECLARE MULTICHAR SYMBOLS /////

Multichar_Symbols

! *UPPER symbols (Tags)*

+
+1P
+2P
+3P
+ABS
+AUG
+COND
+CONJ
+CONSUETUD
+EMPH
+FEM
+FUT
+IMPERS
+IPF
+IMP
+IND
+MASC
+NEUT
+PASS
+PAST
+PL
+PRON
+PRS
+PRT
+PV1
+PV2
+PV3
+PV4
+REL
+RUL
+SG
+SUBJ
+SUBST
+VROOT
+W1
+W2a

! *LOWER symbols*

^D
^F
^M
```

```
^N
^S
^U
^V
^Vemph1SG
^Vemph2SG
^Vemph3P
^UNKNOWN

! *Flags*

@P.W2a.ON@
@R.W2a.ON@
@D.PV@
@D.PART.NO@
@R.PV.AD@
@R.PV.ARE@
@R.PV.COM@
@R.PV.FO@
@R.PV.IMBI@
@R.PV.TO@
@R.PV.SV@

!\\\\\ BEGIN CONTINUATION CLASSES /////

!\\\\\ STEMS /////

!*** Root = start ***

LEXICON Root
@D.PV@          simpleStems;
@D.PART.NO@     compoundStems;
                substV;

!*** simple vs. compound ***

LEXICON simpleStems
    simpleW1;
    simpleW2a;
! continuation classes for strong types can be added later

LEXICON compoundStems
    compoundW1;
    compoundW2a;
! continuation classes for strong types can be added later

!*** Shell script inserts stems, maintained in separate files,
    for <PLACEHOLDERS> ***
```

```

LEXICON simpleW1
<INSERT SIMPLE W1 STEMS>

LEXICON simpleW2a
<INSERT SIMPLE W2a STEMS>

LEXICON compoundW1
<INSERT COMPOUND W1 STEMS>

LEXICON compoundW2a
<INSERT COMPOUND W2a STEMS>

!*** Weak stem formation ***

LEXICON W1
+W1:ā   weakStemFormation;

LEXICON W2a
@P.W2a.ON@+W2a:ī    weakStemFormation;

LEXICON weakStemFormation
+PRS+IND:0      weakPresIndEndings;
+IMP:0          weakImpEndings;
+IPF:0          secEndings;
+PRS+SUBJ:0     aEndings;
+PAST+SUBJ:0    secEndings;
+FUT:^F         aEndings;
+COND:^F        secEndings;
+PRT:^S         sPretEndings;
+PRT+PASS:θ     pretPassEndings;

!\\\\\ ENDINGS /////

LEXICON weakPresIndEndings
+ABS+1P+SG:^M'              Emph1sg;
@R.W2a.ON@+ABS+1P+SG:u      Emph1sg;
+ABS+2P+SG:i                Emph2sg;
+ABS+3P+SG:θ'               suffAbs3sg;
+ABS+3P+SG+REL:s            Emph3sg;
+ABS+1P+PL:^M'i             Emph1pl;
+ABS+1P+PL+REL:^M'e         Emph1pl;
+ABS+2P+PL:θ'e              Emph2pl;
+ABS+3P+PL:t'               Emph3pl;
+ABS+3P+PL+REL:^V^D'e       Emph3pl;
+CONJ+1P+SG:^M'             Emph;
@R.W2a.ON@+CONJ+1P+SG:u     Emph;
+CONJ+2P+SG:i               Emph;
```

```
+CONJ+3P+SG:0               Emph;
+CONJ+1P+PL:µ               Emph;
+CONJ+2P+PL:θ'              Emph;
+CONJ+3P+PL:t               Emph;
+PASS:0                     pass1Endings;

LEXICON weakImpEndings
+CONJ+1P+SG:^M'             Emph;
@R.W2a.ON@+CONJ+1P+SG:u     Emph;
+CONJ+2P+SG:∅               Emph;
+CONJ+3P+SG:θ               Emph;
+CONJ+1P+PL:µ               Emph;
+CONJ+2P+PL:θ'              Emph;
+CONJ+3P+PL:t               Emph;
+PASS:0                     pass1Endings;

LEXICON aEndings
+ABS+1P+SG:a            Emph1sg;
+ABS+2P+SG:e            Emph2sg;
+ABS+3P+SG:θ'           suffAbs3sg;
+ABS+3P+SG+REL:s        Emph3sg;
+ABS+1P+PL:^M'i         Emph1pl;
+ABS+1P+PL+REL:^M'e     Emph1pl;
+ABS+2P+PL:θ'e          Emph2pl;
+ABS+3P+PL:t'           Emph3pl;
+ABS+3P+PL+REL:^V^D'e   Emph3pl;
+CONJ+1P+SG:∅           Emph;
+CONJ+2P+SG:e           Emph;
+CONJ+3P+SG:a           Emph;
+CONJ+1P+PL:µ           Emph;
+CONJ+2P+PL:θ'          Emph;
+CONJ+3P+PL:t           Emph;
+PASS:0                 pass1Endings;

LEXICON sPretEndings
+ABS+1P+SG:u            Emph1sg;
+ABS+2P+SG:'i           Emph2sg;
+ABS+3P+SG:'            suffAbs3sg;
+ABS+3P+SG+REL:0        Emph3sg;
+ABS+1P+PL:ə^M'i        Emph1pl;
+ABS+1P+PL+REL:ə^M'e    Emph1pl;
+ABS+2P+PL:^UNKNOWN     Emph2pl;
+ABS+3P+PL:ət'          Emph3pl;
+ABS+3P+PL+REL:ət'e     Emph3pl;
+CONJ+1P+SG:^U          Emph;
+CONJ+2P+SG:'           Emph;
+CONJ+3P+SG:∅           Emph;
+CONJ+1P+PL:əµ          Emph;
```

```
+CONJ+2P+PL:əθ'         Emph;
+CONJ+3P+PL:ət          Emph;

LEXICON pass1Endings
+ABS+3P+SG:θ'ər'        Emph3sg;
+ABS+3P+SG+REL:θ'ər     Emph3sg;
+ABS+3P+PL:t'ər'        Emph3pl;
+ABS+3P+PL+REL:^V^Dər   Emph3pl;
+CONJ+3P+SG:θ'ər        Emph;
+CONJ+3P+PL:^V^Dər      Emph3pl;

LEXICON pretPassEndings
+ABS+3P+SG:e        Emph3sg;
+ABS+3P+SG+REL:e    Emph3sg;
+ABS+3P+PL:i        Emph3pl;
+ABS+3P+PL+REL:i    Emph3pl;
+CONJ+3P+SG:0       Emph;
+CONJ+3P+PL:a       Emph3pl;

LEXICON secEndings
+CONJ+1P+SG:^N'         Emph;
+CONJ+2P+SG:θa          Emph;
+CONJ+3P+SG:θ           Emph;
+PASS+CONJ+3P+SG:θ'e    Emph;
+CONJ+1P+PL:^M'əs'      Emph;
+CONJ+2P+PL:θ'e         Emph;
+CONJ+3P+PL:t'əs'       Emph;
+PASS+CONJ+3P+PL:t'əs'  Emph;

!\\\\\ SUBSTANTIVE VERB /////

LEXICON substV
@R.PV.SV@+tá+VROOT+SUBST+PRS+IND:0      Tá;
@D.PV@+tá+VROOT+SUBST+PRS+IND:0         Tá;
@D.PV@+fil+VROOT+SUBST+PRS+IND:0        Fil;
@D.PV@                                  Bí;
@D.PV@                                  TápresWithSuffPron;
@D.PV@                                  BípretWithSuffPron;

LEXICON Tá
+CONJ+1P+SG:táu                     Emph1sg;
+CONJ+1P+SG:tó                      Emph2sg;
+CONJ+3P+SG:tá                      Emph3sg;
@D.PV@+IMPERS+CONJ+3P+SG:táthar     #;
+CONJ+1P+PL:taam                    Emph1pl;
+CONJ+2P+PL:taid                    Emph2pl;
+CONJ+3P+PL:taat                    Emph3pl;

```

```
LEXICON Fil
+ABS+3P+SG:fil          #;
+ABS+3P+SG+REL:fil      #;
+ABS+3P+SG+REL:file     #;
+CONJ+3P+SG:fil         #;

LEXICON Bí
    SVconsuetud;
    SVimpf;
    SVpret;
    SVfut;
    SVcond;
    SVimp;
    SVpresSubj;
    SVpastSubj;

LEXICON SVconsuetud
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+1P+SG:biuu
    Emph1sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+3P+SG:biid
    Emph3sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+3P+SG+REL:biis
    Emph3sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+3P+SG+REL:bís
    Emph3sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+IMPERS+ABS+3P+SG:bíthir   #;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+1P+PL:bímmi
    Emph1pl;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+1P+PL+REL:bímme
    Emph1pl;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+3P+PL:biit
    Emph3pl;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+ABS+3P+PL+REL:bíte
    Emph3pl;

+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+1P+SG:bíu
        Emph1sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+2P+SG:bí
        Emph2sg;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+3P+SG:bí
        Emph;
+ro+AUG+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+3P+SG:rubai
        Emph;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+IMPERS+CONJ+3P+SG:bíther
        #;
+ro+AUG+bí+VROOT+SUBST+PRS+IND+CONSUETUD+IMPERS+CONJ+3P+SG:rubthar
    #;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+1P+PL:biam
```

```
        Emph1pl;
+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+3P+PL:biat
        Emph3pl;
+ro+AUG+bí+VROOT+SUBST+PRS+IND+CONSUETUD+CONJ+3P+PL:rubat
        Emph3pl;

LEXICON SVimpf
+bí+VROOT+SUBST+IPF+CONJ+1P+SG:biinn            Emph1sg;
+bí+VROOT+SUBST+IPF+CONJ+3P+SG:bíth             Emph;
+bí+VROOT+SUBST+IPF+IMPERS+CONJ+3P+SG:bíthe     #;
+bí+VROOT+SUBST+IPF+CONJ+1P+PL:bimmis           Emph1pl;
+bí+VROOT+SUBST+IPF+CONJ+3P+PL:bítis            Emph3pl;

LEXICON SVpret
+bí+VROOT+SUBST+PRT+ABS+1P+SG:bá                Emph1sg;
+bí+VROOT+SUBST+PRT+ABS+2P+SG:bá                Emph2sg;
+bí+VROOT+SUBST+PRT+ABS+3P+SG:boí               Emph3sg;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+REL:boíe          Emph3sg;
+bí+VROOT+SUBST+PRT+IMPERS+ABS+3P+SG:bothae     #;
+bí+VROOT+SUBST+PRT+ABS+1P+PL:bámmar            Emph1pl;
+bí+VROOT+SUBST+PRT+ABS+3P+PL:bátar             Emph3pl;

+bí+VROOT+SUBST+PRT+CONJ+1P+SG:bá               Emph1sg;
+bí+VROOT+SUBST+PRT+CONJ+2P+SG:bá               Emph2sg;
+bí+VROOT+SUBST+PRT+CONJ+3P+SG:boí              Emph;
+bí+VROOT+SUBST+PRT+IMPERS+CONJ+3P+SG:both      #;
+bí+VROOT+SUBST+PRT+CONJ+1P+PL:bámmar           Emph1pl;
+bí+VROOT+SUBST+PRT+CONJ+2P+PL:báid             Emph2pl;
+bí+VROOT+SUBST+PRT+CONJ+3P+PL:bátar            Emph3pl;

+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+1P+SG:roba              Emph1sg;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+1P+SG:raba              Emph1sg;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+2P+SG:raba              Emph2sg;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+3P+SG:robae             Emph;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+3P+SG:rabae             Emph;
+ro+AUG+bí+VROOT+SUBST+PRT+IMPERS+CONJ+3P+SG:robad      #;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+1P+PL:roba^Vmmar        Emph1pl;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+2P+PL:robaid            Emph2pl;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+3P+PL:robatar           Emph3pl;
+ro+AUG+bí+VROOT+SUBST+PRT+CONJ+3P+PL:rabatar           Emph3pl;

LEXICON SVfut
+bí+VROOT+SUBST+FUT+ABS+1P+SG:bia               Emph1sg;
+bí+VROOT+SUBST+FUT+ABS+2P+SG:bie               Emph2sg;
+bí+VROOT+SUBST+FUT+ABS+3P+SG:bieid             Emph3sg;
+bí+VROOT+SUBST+FUT+ABS+3P+SG:bied              Emph3sg;
+bí+VROOT+SUBST+FUT+ABS+3P+SG+REL:bias          Emph3sg;
+bí+VROOT+SUBST+FUT+IMPERS+ABS+3P+SG:bethir     #;
```

```
+bí+VROOT+SUBST+FUT+ABS+1P+PL:bemmi             Emph1pl;
+bí+VROOT+SUBST+FUT+ABS+2P+PL:bethe             Emph2pl;
+bí+VROOT+SUBST+FUT+ABS+3P+PL:bieit             Emph3pl;
+bí+VROOT+SUBST+FUT+ABS+3P+PL+REL:bete          Emph3pl;

+bí+VROOT+SUBST+FUT+CONJ+3P+SG:bia      Emph;
+bí+VROOT+SUBST+FUT+CONJ+1P+PL:biam     Emph1pl;
+bí+VROOT+SUBST+FUT+CONJ+2P+PL:bieid    Emph2pl;
+bí+VROOT+SUBST+FUT+CONJ+2P+PL:bied     Emph2pl;
+bí+VROOT+SUBST+FUT+CONJ+3P+PL:biat     Emph3pl;

LEXICON SVcond
+bí+VROOT+SUBST+COND+CONJ+1P+SG:beinn   Emph1sg;
+bí+VROOT+SUBST+COND+CONJ+3P+SG:biad    Emph;
+bí+VROOT+SUBST+COND+CONJ+1P+PL:bemmis  Emph1pl;
+bí+VROOT+SUBST+COND+CONJ+3P+PL:betis   Emph3pl;

LEXICON SVimp
+bí+VROOT+SUBST+IMP+CONJ+2P+SG:bí       Emph1sg;
+bí+VROOT+SUBST+IMP+CONJ+3P+SG:biid     Emph2sg;
+bí+VROOT+SUBST+IMP+CONJ+3P+SG:bíth     Emph;
+bí+VROOT+SUBST+IMP+CONJ+2P+PL:biid     Emph2pl;
+bí+VROOT+SUBST+IMP+CONJ+2P+PL:bíth     Emph2pl;
+bí+VROOT+SUBST+IMP+CONJ+3P+PL:biat     Emph3pl;

LEXICON SVpresSubj
+bí+VROOT+SUBST+PRS+SUBJ+ABS+1P+SG:béo              Emph1sg;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+1P+SG:béu              Emph1sg;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+2P+SG:bé               Emph2sg;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+2P+SG:bee              Emph2sg;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+3P+SG:beiθ             Emph3sg;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+3P+SG+REL:bes          Emph3sg;
+bí+VROOT+SUBST+PRS+SUBJ+IMPERS+ABS+3P+SG:bethir    #;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+1P+PL:bemmi            Emph1pl;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+2P+PL:bethe            Emph2pl;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+3P+PL:beit             Emph3pl;
+bí+VROOT+SUBST+PRS+SUBJ+ABS+3P+PL+REL:bete         Emph3pl;

+bí+VROOT+SUBST+PRS+SUBJ+CONJ+1P+SG:béo     Emph1sg;
+bí+VROOT+SUBST+PRS+SUBJ+CONJ+2P+SG:bé      Emph2sg;
+bí+VROOT+SUBST+PRS+SUBJ+CONJ+3P+SG:bé      Emph;
+bí+VROOT+SUBST+PRS+SUBJ+CONJ+1P+PL:bem     Emph1pl;
+bí+VROOT+SUBST+PRS+SUBJ+CONJ+2P+PL:beid    Emph2pl;
+bí+VROOT+SUBST+PRS+SUBJ+CONJ+3P+PL:bet     Emph3pl;

+ro+AUG+bí+VROOT+SUBST+PRS+SUBJ+CONJ+3P+SG:roib     Emph;
+ro+AUG+bí+VROOT+SUBST+PRS+SUBJ+CONJ+1P+PL:robam    Emph1pl;
+ro+AUG+bí+VROOT+SUBST+PRS+SUBJ+CONJ+2P+PL:robith   Emph2pl;
```

```
+ro+AUG+bí+VROOT+SUBST+PRS+SUBJ+CONJ+3P+PL:robat    Emph3pl;

LEXICON SVpastSubj
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+1P+SG:beinn          Emph1sg;
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+2P+SG:betha          Emph2sg;
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+3P+SG:beθ            Emph3sg;
+bí+VROOT+SUBST+PAST+SUBJ+IMPERS+CONJ+3P+SG:bethe   #;
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+1P+PL:bemmis         Emph1pl;
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+2P+PL:bethe          Emph2pl;
+bí+VROOT+SUBST+PAST+SUBJ+CONJ+3P+PL:betis          Emph3pl;

+ro+AUG+bí+VROOT+SUBST+PAST+SUBJ+CONJ+3P+SG:robad       Emph;
+ro+AUG+bí+VROOT+SUBST+PAST+SUBJ+CONJ+3P+PL:roibtis     Emph3pl;

LEXICON TápresWithSuffPron
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+1P+SG:táthum         #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+2P+SG:táthut         #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+SG+MASC:táithi    #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+SG+MASC:táthai    #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+SG+NEUT:táithi    #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+SG+NEUT:táthai    #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+SG+FEM:táthus     #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+1P+PL:táthunn        #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+1P+PL:táithiunn      #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+2P+PL:táthuib        #;
+tá+VROOT+SUBST+PRS+IND+ABS+3P+SG+PRON+3P+PL:táthus         #;

LEXICON BípretWithSuffPron
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+1P+SG:baíthum        #;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+2P+SG:baíthut        #;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+3P+SG+MASC:baíthi    #;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+3P+SG+NEUT:baíthi    #;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+3P+SG+FEM:boíthus    #;
+bí+VROOT+SUBST+PRT+ABS+3P+SG+PRON+3P+PL:boíthus        #;

!!\\\\\ SUFFIXES /////

LEXICON Emph
        #;
0:-     Emph2;

LEXICON Emph2
+EMPH+1P+SG:s^Vemph1SG          #;
+EMPH+2P+SG:s^Vemph2SG          #;
+EMPH+3P+SG+MASC:s^Vemph3Pm     #;
+EMPH+3P+SG+NEUT:s^Vemph3Pm     #;
+EMPH+3P+SG+FEM:si              #;
+EMPH+1P+PL:ni                  #;
```

```
+EMPH+2P+PL:si                  #;
+EMPH+3P+PL:s^Vemph3Pm          #;

LEXICON suffAbs3sg
                        Emph3sg;
+PRON+1P+SG:um          #;
+PRON+2P+SG:ut          #;
+PRON+3P+SG+MASC:i      #;
+PRON+3P+SG+NEUT:i      #;
+PRON+3P+SG+FEM:us      #;
+PRON+1P+PL:unn         #;
+PRON+2P+PL:uib         #;
+PRON+3P+PL:us          #;

LEXICON Emph1sg
                            #;
+EMPH+1P+SG:-s^Vemph1SG     #;

LEXICON Emph2sg
                            #;
+EMPH+2P+SG:-s^Vemph2SG     #;

LEXICON Emph3sg
                                #;
+EMPH+3P+SG+MASC:-s^Vemph3Pm    #;
+EMPH+3P+SG+NEUT:-s^Vemph3Pm    #;
+EMPH+3P+SG+FEM:-si             #;

LEXICON Emph1pl
                    #;
+EMPH+1P+PL:-ni     #;

LEXICON Emph2pl
                    #;
+EMPH+2P+PL:-si     #;

LEXICON Emph3pl
                            #;
+EMPH+3P+PL:-s^Vemph3Pm     #;
```

C.1.4 tbf.lexc

```
!***** tbf.lexc *****
! Th. Fransen, 18/08/19
! A selection of nouns, proper names, function words as well as defective "ol" taken from Táin Bó Fraích (Meid 1974)

```

```

!\\\\\ DECLARE MULTICHAR SYMBOLS /////

Multichar_Symbols

! *UPPER symbols (Tags)*

+1P
+2P
+3P
+ACC
+ADV
+ART
+ACC
+CONJUNCTION
+DAT
+DEFECT
+FEM
+GEN
+H
+LEN
+MASC
+NAS
+NEUT
+NOM
+NOUN
+PART
+PL
+POSS
+PREP
+PRON
+PROP
+SG
+TEMP
+VERB
+VOC

!\\\\\ BEGIN CONTINUATION CLASSES /////

!*** Root = start ***

LEXICON Root
    Conjunction;
    Article;
    Preposition;
    Proper;
    Particle;
    possPronoun;
    Noun;
    Adverb;
    Verb;

LEXICON Conjunction
ocus+CONJUNCTION:&      #;

LEXICON Article
a+ART+NEUT+SG+ACC+NAS:a     #;
a+ART+NEUT+SG+NOM+NAS:a     #;
in+ART:in                   #;
in+ART+FEM+GEN+SG:na        #;
in+ART+GEN+PL:na            #;

LEXICON Preposition
a+PREP+DAT:a                        #;
a+PREP+DAT+PRON+3P+SG+NEUT:ass      #;
do+PREP+LEN:do                      #;
i+PREP+NAS:i                        #;

LEXICON Proper
Aillil+PROP+NOUN+SG+GEN:Ailella         #;
Aillil+PROP+NOUN+SG+ACC:Ailill          #;
Aillil+PROP+NOUN+SG+DAT:Aillil          #;
Aillil+PROP+NOUN+SG+NOM:Ailill          #;
Boind+PROP+NOUN+SG+ACC:Bóind            #;
Boind+PROP+NOUN+SG+DAT:Boind            #;
Boind+PROP+NOUN+SG+NOM:Boind            #;
Boind+PROP+NOUN+SG+GEN:Bóinni           #;
Cernach+PROP+NOUN+ACC:Cernach           #;
Cernach+PROP+NOUN+NOM:Cernach           #;
Cernach+PROP+NOUN+GEN:Chernaig          #;
Conall+PROP+NOUN+GEN:Conaill            #;
Conall+PROP+NOUN+ACC:Conall             #;
Conall+PROP+NOUN+NOM:Conall             #;
Findabair+PROP+NOUN+ACC:Findabair       #;
Findabair+PROP+NOUN+DAT:Findabair       #;
Findabair+PROP+NOUN+NOM:Findabair       #;
Findabair+PROP+NOUN+ACC:Finndabair      #;
Findabair+PROP+NOUN+DAT:Finndabair      #;
Findabair+PROP+NOUN+NOM:Finndabair      #;
Fróech+PROP+NOUN+SG+NOM:Fráech          #;
Fróech+PROP+NOUN+SG+GEN:Fraích          #;
Fróech+PROP+NOUN+SG+ACC:Fróech          #;
Fróech+PROP+NOUN+SG+NOM:Fróech          #;
Ériu+GEN:Hérenn                         #;
Ériu+NOM:Hériu                          #;
Medb+PROP+NOUN+GEN:Medba                #;
```

```
Medb+PROP+NOUN+NOM:Medb                 #;
Medb+PROP+NOUN+VOC:Medb                 #;
Medb+PROP+NOUN+ACC:Meidb                #;
Medb+PROP+NOUN+DAT:Meidb                #;
Fidach?+PROP+NOUN+GEN:Idaith            #;

LEXICON Particle
PART+VOC+LEN:a  #;

LEXICON possPronoun
do+POSS+PRON+2P+SG+LEN:do           #;
PRON+POSS+3P+SG+MASC+GEN+LEN:a      #;
PRON+POSS+3P+SG+MASC+LEN:a          #;
PRON+POSS+3P+SG+FEM+GEN+H:a         #;
PRON+POSS+3P+SG+FEM+H:a             #;
PRON+POSS+3P+SG+NEUT+GEN+LEN:a      #;
PRON+POSS+3P+SG+NEUT+LEN:a          #;
PRON+POSS+3P+PL+GEN+NAS:a           #;
PRON+POSS+3P+PL+NAS:a               #;

LEXICON Noun
bó+NOUN+PL+GEN:bó       #;
bó+NOUN+SG+NOM:bó       #;
macc+PL+ACC:maccu       #;
macc+PL+GEN:mac         #;
macc+SG+NOM:mac         #;
macc+PL+NOM:maicc       #;
macc+SG+GEN:maicc       #;
macc+SG+GEN:maic        #;
macc+PL+NOM:meicc       #;
táin+NOUN+SG+NOM:táin   #;

LEXICON Adverb
íarum+ADV+TEMP:íarum    #;

LEXICON Verb
ol+VERB+DEFECT:ol   #;
```

C.2 Rules (.rule)

C.2.1 1_se_filters.rule

```
#***** 1_se_filters.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Upper-level filters to remove incompatible tag combinations
    from the network
```

```

regex [
# No preceding elements with absolute endings
# This also excludes prototonic bases (+PV1 can only be protot in se.lexc)
~[ $[ ["+PV1"|"+AUG"|"+IMP"] ?* "+ABS"] ] .o.

# complement of @D.PV@ = @R.PV.X = deut
~[ ~$["@D.PV@"] & $["+ABS"] ] .o.

# no augment with the imperative
~[ $[ "+AUG" ?* "+IMP" ]]
] ;
```

C.2.2 2_se_phon_non_pres_stem_form.rule

```
#***** 2_se_phon_non_pres_stem_form.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Replace rules regarding non-present weak stem formation (f-fut, s-pret)

regex [
# e.g. marbā-iF -> marbi^F, marbā^F, léicī^F -> léici^F
[ ā (->) i , ī -> i || _ "^F" ] .o.

# marbā^FØ -> marbu^FØ, marbi^FØ -> marbiu^FØ, léici^FØ -> léiciu^FØ
[ ā -> u , i -> {iu} || _ "^F" Ø [.#. | "-" ] ] .o.

[ [..] -> ə || "^F" _ Cons ] .o.
["^S" -> 0 || _ Ø ] .o.

# *léicis wrong pret rel. 3sg?
# make sure ī becomes i before s-stem consonant (not marked palatal), otherwise lowered
[ī -> i || _ "^S" ] .o.
[ "^S" "^U" -> u "^S" ] #[ "^S" "^U" -> [u|a] "^S" ]
] ;
```

C.2.3 3_se_phon_lowering.rule

```
#***** 3_se_phon_lowering.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Lowering of stem vowel

# lowering e.g. dep./conj. 1pl léicīm (m non-pal) > léicem, léicīa > léicea (subj. abs 1sg/conj 3sg)
# this rule before palatalisation or else e.g. 1pl. léicīm > léic'īm' (m wrongly pal.) > **léicim.
# ^V in rule context for subsequent optional syncope; e.g. prs. ind./subj. pass. 3pl. conj./abs. rel. léicī^V^Dər > léice^V^Dər
# also secondary ending 3sg.
regex [ ī -> e || _ [ ("^V") nonPalCons+ [ Vow | .#. | "-"] ] | a ] ;
```

C.2.4 4_se_phon_pal.rule

```
##### 4_se_phon_pal.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Add palatalisation markers

regex [
[..] -> ' || nonPalCons+ _ nonPalCons* frontVow ,
[i|ī] nonPalCons+ _ nonPalCons* [ Vow | .#. | "-" ]
] ;
```

C.2.5 5_se_phon_del_stem_vow.rule

```
##### 5_se_phon_del_stem_vow.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Delete W1/W2a stem vowels before vowels and where there is no end vowel

# Rules should apply after e.g. palatalisation otherwise endings such as pret. conj. 3sg tarlaic become tarlac.
# stem vowel in tarələcī, that points at palatal auslaut, is needed for palat. rule.

# e.g. imp 2sg marbā-∅, subj abs 1 sg marbā-a
# pres.ind. conj 2sg léicī-i, imp 2sg léicī-∅
regex [
[ā -> 0 || _ a|u|∅ ] .o. # e.g.
[ī -> 0 || _ e|i|∅ ] .o. # e.g.
[∅ -> 0 ]
] ;
```

C.2.6 6_se_phon_syncope.rule

```
#***** 6_se_phon_syncope.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Syncope

# mark syncope
regex [
Vow+ @-> "[" ... "]" ||
.#. Cons* [ Vow+ Cons+       |
          [Vow+ Cons+]^3     |
          [Vow+ Cons+]^5 ] _ Cons+ Vow
] .o.

# phonotactic exceptions on syncope
# some consonants have (underlying) symbols!
# ·ro-m[a]rb-     (mrb)
# ·ad-r[o-e]ll- (drl), roibr[i]ssimm (brs) (hypothethical; not part of current FST)
[ "[" -> 0 , "]" -> 0 ||
_ (Vow+ "]") r (') Cons ,
Cons r (') ("[" Vow+) _ (Vow+ "]") [l|s|"^S"]
] .o.

# optional syncope with e.g. abs. rel. 3pl. using ^V
# e.g. prs. ind. 3pl. rel. marb[ā^V]^D'e AND marb[ā^V^D'e
# Rule context with preceding Vow does not interfere with f-future (taking a- and pass1 endings), e.g. marb[ā]^F^V^D'e
[ "]" (->) 0 || "[" Vow "^V" _ ] .o.
["^V" -> 0 || Vow _ ] .o.
["^V" -> ə ] .o.

# delete syncope markers
# e.g. marb[ā]^D'e > marb^D'e
[ "[" Vow+ "]" -> 0 ]  .o.

# delete "[" with optional syncope which has remained up until now
# e.g. marb[ā^D'e > marbā^D'e
[ "[" -> 0 ] ;
```

C.2.7 7_se_phon_cons_qual_assim.rule

```
#***** 7_se_phon_cons_qual_assim.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Consonant quality assimilation of adjacent consonant clusters
    after syncopated vowels

regex [
[ [..] -> ' || palCons nonPalCons+ _ nonPalCons* Vow ] .o.
[ ' -> 0 || nonPalCons palCons* nonPalCons _ ]
] ;
```

C.2.8 8_se_phon_stem_vow.rule

```
#***** 8_se_phon_stem_vow.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Vowel rewrite rules as part of the stem

regex [
# create more fine-grained rules later with pal./ non.pal
    (<sync) and a/i + F-future stem cons.
[i -> {ai} || nonPalCons _ "^F" ] .o.
[ā -> 0 || palCons _ i ] .o.

# e.g. protot. ad'll'ā^M' > aidlim, ad'l'l'ā^F'e > aidlibe
[ā -> i || palCons _ palCons ] .o.

# e.g. ad'l'l'āθ > aidled, ad'l'l'ā^Sət> **aidlesat (= adallsat)
[ā -> e || palCons _ nonPalCons ] .o.
[ā -> {ea} || palCons _ ] .o.

# e.g. 3pl. rel. marbaite
[ [..] -> i || ā _ palCons ] .o.
[ā -> a ] .o.
[ī -> a || nonPalCons _ nonPalCons ] .o.

# e.g. l'éic'ī-s'
[ī -> i]
] ;
```

C.2.9 9_se_orth_vow_pal_cons.rule

```
#***** 9_se_orth_vow_pal_cons.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# "i" after back vowel/e/é and pal. consonant

# e.g. ad-elliub > ad-eilliub
regex [ [..] -> i || backVow|e|é _ palCons ] ;
```

C.2.10 10_se_orth_end_vow.rule

```
#***** 10_se_orth_end_vow.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Vowel rewrite rules in post-tonic syllables

# Open syllables

regex [
# e.g. marbtae (pres. 3pl. rel.)
[ [..] -> a || nonPalCons _ [e|i] [.#. | "-" ] ] .o.

# e.g. mairbfea (fut.)
[ [..] -> e || palCons _ a [ .#. | "-" ] ] .o.

# Closed syllables
# schwa
[ ə -> i    || palCons _ palCons ]      .o.
[ ə -> e    || palCons _ nonPalCons ]   .o.
[ ə -> {ai} || nonPalCons _ palCons ]   .o.
[ ə -> a    || nonPalCons _ nonPalCons ]
] .o.

# Open and closed
# palatal consonant + u
# e.g. léicsu -> léicsiu (pret. 1sg. conj)
# suff. pronouns such as léicthunn -> léicthiunn etc.
[ [..] -> i || palCons _ u ] ;
```

C.2.11 11_se_vow_emph.rule

```
#***** 11_se_vow_emph.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Rewrite abstract vowel symbols with emph. particles

regex [
# palatal
[ "^Vemph1SG" -> e (a) , "^Vemph2SG" -> {iu}  , "^Vemph3P" -> e (o) | {iu} || [ frontVow | palCons ] "-" s _ ] .o.

# non-palatal
[ "^Vemph1SG" -> a , "^Vemph2SG" -> o|u , "^Vemph3P" -> a|o
|| [ backVow | nonPalCons ] "-" s _ ]
] ;
```

C.2.12 12_se_del_pal_markers.rule

```
#***** 12_se_del_pal_markers.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script

regex [' -> 0 ] ;
```

C.2.13 13_se_phon_orth_cons.rule

```
#***** 13_se_phon_orth_cons.rule *****
# Th. Fransen, 16/08/19
# Invoked as part of se_1_bare.script
# Rewrite phonological and abstract orthographic consonant symbols

regex [
# consecutive s's
["^S" -> 0 || {ss} _ ] .o.      # bris(s)-^Sis
["^S" -> [ 0 | s ] || s _ ] .o.
[{ss} (->) s] .o.
["^S" -> s] .o.

# Delenition
[ θ -> t || d|l|n|s|t _ ] .o.
[ d -> 0 , t (->) 0 || _ t  ] .o.

# voicing / devoicing
[θ -> {th} || Cons _ ] .o.
[θ -> {th}|d ] .o.

# protot. aid(e)l- ( < ad-ell)
[ l -> 0 || d _ l ] .o.

["^D" -> t || Vow _ ] .o.
["^D" -> d|t ] .o.
["^M" -> m || Cons _ Vow ] .o.
["^M" -> m | {mm} ] .o.
["^N" -> {nn}] .o.
[μ -> m] .o.
[b (->) 0 || _ "^F" ] .o.
["^F" -> b || _ .#. | "-" ] .o.
["^F" (->) b || Vow _ ] .o.
["^F" -> f]
] ;
```

C.2.14 procl_se_filter_1_aug.rule

```
#***** procl_se_filter_1_aug.rule *****
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out forms incompatible with the augment

regex [
~[ $["+AUG" ?* "+AUG"] ] .o.
~[ $["+AUG" ?* "+IMP"] ] .o.
~[ $["+AUG" ?* "+PROCL_JUNCT" ?* "+PV1"] ] .o.
~[ $[ "+AUG" ?* [{tá}|{fil}] "+VROOT" "+SUBST"] ]
] ;
```

C.2.15 procl_se_filter_2_no.rule

```
#***** procl_se_filter_2_no.rule *****
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out tags incompatible with "no"

regex [
# e.g. **no·reilic
~[ $[ {no} "+CONJ_PART" ?* "+AUG" ] ] .o.

# "No" (non-rel.) without inf. pronoun only with secondary
    endings (i.e. not with tenses/moods below)
~[ $[{no} "+CONJ_PART"] & ~$["+REL"] & ~$["+PRON"] &
    $["+PRS"|"+IMP"|"+FUT"|"+PRT"] ] .o.

# Relative "no" forms WITHOUT inf. pron. are restricted to those
    person/number forms that do not have a special absolute rel.,
    e.g. prs. ind. 1pl. rel. **no·léicem (> léicme), but 1pl.
    secondary end. (invariably conj.) no·léic(fi)mis (both main
    and relative)
~[ $[{no} "+CONJ_PART" ?* "+REL"] & ~$["+PRON"] & $["+CONJ"
    ["+3P" | "+1P" "+PL"] ] & $["+PRS"|"+IMP"|"+FUT"|"+PRT"] ]
] ;
```

C.2.16 procl_se_filter_3_ipv.rule

```
#***** procl_se_filter_3_ipv.rule *****
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out proclitics (+ relative) incompatible with the
    imperative

regex [
# No relative with the imperative
~[ $["+REL" ?* "+IMP"] ] .o.

# An imperative particle cannot go with anything other than an
    imperative form
~[ $["+CONJ_PART" "+IMP" "+NEG"] &  ~$["+IMP" ("+PASS") "+CONJ"]
    ] .o.

# No procl. prefixes other than the negative imperative particle
    or "no" with imperatives
~[ $["+CONJ_PART"] & ~$["+IMP" "+NEG" | {no} "+CONJ_PART"] &
    $["+IMP" ("+PASS") "+CONJ"] ] .o.

# Deuterotonic imperative form not allowed except when infixed
    pronoun present
~[ $["+PV1" ?* "+PROCL_JUNCT"] & ~$["+PRON"] & $["+IMP"
    ("+PASS") "+CONJ"] ]
] ;
```

C.2.17 procl_se_filter_4_pass.rule

```
#***** procl_se_filter_4_pass.rule *****
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out proclitics incompatible with passive endings

regex [
# no 3pl. pass. with inf. pron., e.g. **nob·marbtar,
    **don·léicfiter
~[ $[ [{no} "+CONJ_PART" | "+PRON"] ?* "+PASS" "+CONJ" "+3P"
    "+PL"] ] .o.

# no 3sg./pl. inf. pron. with pass. (sg.), e.g.
    **na/ra/da·léicther
~[ $[ ["+CONJ_PART"|"+PV1"|"+AUG"] ?* "+PRON" (?) "+3P" ?*
    "+PASS" "+CONJ" "+3P" "+SG"] ]
] ;
```

C.2.18 procl_se_filter_5_sv.rule

```
#***** procl_se_filter_5_sv.rule
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out tags incompatible with the substantive verb
# GOI 476--483

regex [
# Pronouns are not allowed with impersonals
~[ $[ ["+CONJ_PART"|"+AUG"] ?* ["+PRON"] ?* [{tá}|{bí}] "+VROOT"
    "+SUBST" ?* "+IMPERS"]] .o.

# e.g. at·taam but **ní·taam (nín·fil), only dependent ·tá
    exists (but see next restriction)
~[ $["+CONJ_PART" ?* {tá} "+VROOT" "+SUBST"] & ~$["+CONJ" "+3P"
    "+SG"] ] .o.

# ·tá, when preceded by a conj. part., needs (dative) pronoun
    (**ní·tá).
# currently nom·tá etc allowed (alongside táthum)
# Currently e.g. no(n)dom·t(h)á (rel. ) ''that I have'' allowed
    but check grammaticality len./nas. relative in the context of
    subject/obj. antecedent

~[ $["+CONJ_PART" ?* {tá} "+VROOT" "+SUBST"] & ~$["+PRON"]
] .o.

# 'fil' (3sg) can occur without infix.pron., e.g. ní·fil 'there
    is not' in contrast to dependent **ní·tá, so different
    restrictions.
# Non-rel constructions like nom·fil 'I am' etc not possible ->
    independ. at·táu etc (but e.g. ním·fil ''I am not'')
# Relative fil(e) e.g. **noda·fil > fil(e) 'that she is' taken
    care of in 'procl_se_filter_no.rule' which states that the
    person/number of the infixed pron. may not appear with 'no'
    if there already is a special absolute relative ending.
# Also: only a leniting relative clause possible with fil

~[ $[ {no} "+CONJ_PART" ?* {fil} "+VROOT" "+SUBST"] & ~$["+REL"] ] .o.
~[ $[ "+REL" "+NAS" ?* {fil} "+VROOT" "+SUBST"] ] .o.

# Bí allows the augment and the only restriction here is its conj. forms. It's like a passive in that the 3rd sg. ending with an infix is used as a dative (is to X). So without an infix all endings are allowed, but with a dative infix only 3sg. ending.
# e.g. ní·boí '(he/she) was, ním·boí 'I had', ní·bámmar 'we were not', but NOT e.g. **ním·bámmar

~[ $[ ["+CONJ_PART"| "+AUG"] ?* "+PRON" ?* {bí} "+VROOT" "+SUBST"] & ~$["+CONJ" "+3P" "+SG"] ]
] ;
```

C.2.19 procl_se_filter_6_mut.rule

```
# ***** procl_se_filter_6_mut.rule
# Th. Fransen, 21/08/19
# Invoked as part of procl_se.script
# Filter out incompatible mutation tags
# Disallow non-matching mutation tags but also the absence of a mutation tag on either side of the procl. juncture, e.g. ro+AUG+PROCL_JUNCT+LEN+bris+VROOT... and ro+AUG+REL+NAS+PROCL_JUNCT+bris+VROOT...
# Note also *AD+PV1+PROCL_JUNCT+LEN+tá:at·thá (no inf. pronoun/relative possible and, consequently, mutations on tá impossible).

regex [
~[ $["+LEN" "+PROCL_JUNCT" \["+LEN"] ] ] .o.
~[ $["+NAS" "+PROCL_JUNCT" \["+NAS"] ] ] .o.
~[ $["+H" "+PROCL_JUNCT" \["+H"] ] ] .o.
~[ $[\["+LEN"] "+PROCL_JUNCT" "+LEN"] ] .o.
~[ $[\["+NAS"] "+PROCL_JUNCT" "+NAS"] ] .o.
~[ $[\["+H"] "+PROCL_JUNCT" "+H"] ]
] ;
```

C.2.20 procl_se_filter_7_emph.rule

```
#***** procl_se_filter_7_emph.rule *****
# Th. Fransen, 20/08/19
# Invoked as part of procl_se.script
# Filter out emphasising particles incompatible with infixed
    pron. and verbal endings
# This file defines temporary lexicons (LEXfilteredTemp2-15)

define LEXfilteredTemp2 [ ~[ ~$["+1P" "+SG" ?* "+EMPH"] &
    $["+EMPH" "+1P" "+SG"] ] .o. LEXfilteredTemp1 ] ;
define LEXfilteredTemp3 [ ~[ ~$["+2P" "+SG" ?* "+EMPH"] &
    $["+EMPH" "+2P" "+SG"] ] .o. LEXfilteredTemp2 ] ;
define LEXfilteredTemp4 [ ~[ ~$["+3P" "+SG" ?* "+EMPH"] &
    $["+EMPH" "+3P" "+SG"] ] .o. LEXfilteredTemp3 ] ;
define LEXfilteredTemp5 [ ~[ ~$["+1P" "+PL" ?* "+EMPH"] &
    $["+EMPH" "+1P" "+PL"] ] .o. LEXfilteredTemp4 ] ;
define LEXfilteredTemp6 [ ~[ ~$["+2P" "+PL" ?* "+EMPH"] &
    $["+EMPH" "+2P" "+PL"] ] .o. LEXfilteredTemp5 ] ;
define LEXfilteredTemp7 [ ~[ ~$["+3P" "+PL" ?* "+EMPH"] &
    $["+EMPH" "+3P" "+PL"] ] .o. LEXfilteredTemp6 ] ;
define LEXfilteredTemp8 [ ~[ $["+MASC" ?* "+EMPH"] & ~
    $[["+ABS"|"+CONJ"] "+3P" "+SG"] & $["+EMPH" "+3P" "+SG"
    ["+NEUT"|"+FEM"]] ] .o. LEXfilteredTemp7 ] ;
define LEXfilteredTemp9 [ ~[ $["+NEUT" ?* "+EMPH"] & ~
    $[["+ABS"|"+CONJ"] "+3P" "+SG"] & $["+EMPH" "+3P" "+SG"
    ["+MASC"|"+FEM"]] ] .o. LEXfilteredTemp8 ] ;
define LEXfilteredTemp10 [ ~[ $["+FEM" ?* "+EMPH"] & ~
    $[["+ABS"|"+CONJ"] "+3P" "+SG"] & $["+EMPH" "+3P" "+SG"
    ["+MASC"|"+NEUT"]] ] .o. LEXfilteredTemp9 ] ;

#\\\\\ PASSIVE AND SV /////

# the emph. part. should refer to the pronoun with the SV and
    passive, which is the subject, not the 3sg ending, i.e. no
    emph. particle 3sg with pass 3sg conj, which only takes
    non-3sg 'subjects' (the incompatible 3sg pronoun infix with
    passives (for which there is an absolute form) already
    filtered out in procl_se_filter_pass.rule)
define LEXfilteredTemp11 [ [ ~[$["+PRON" ?* "+PASS" ?* "+EMPH"
    "+3P" "+SG"]] ] .o. LEXfilteredTemp10 ] ;

# The SV 3sg may take all pronouns and also an emph. part. to
    refer to the 3sg infix, but with other than infixed 3sg the
    emph. part. must refer to the 'subject' infix. pron. and must
    not agree with the 3sg ending.
# As the infixed pron. is the subject of the 3sg ending the
    emph. part. obligatorily agrees with the gender of the infix.
    The ending and emph. particle agreement rule with other verbs
    does not suffice, as there we can have a 3sg ending and any
    3sg. emph. particle, irrespective of the infixed pron.
define LEXfilteredTemp12 [ ~[ $["+PRON" (?) ["+1P"|"+2P"|"+3P"
    "+PL"] ?* "+VROOT" "+SUBST" ?* "+EMPH" "+3P" "+SG"] ] .o.
    LEXfilteredTemp11 ] ;
define LEXfilteredTemp13 [
~[ $["+PRON" (?) "+3P" "+SG" "+MASC" ?* "+VROOT" "+SUBST" ?*
    "+EMPH" "+3P" "+SG" ["+NEUT"|"+FEM"]] ] .o. LEXfilteredTemp12
    ] ;
define LEXfilteredTemp14 [ ~[ $["+PRON" (?) "+3P" "+SG" "+NEUT"
    ?* "+VROOT" "+SUBST" ?* "+EMPH" "+3P" "+SG" ["+MASC"|"+FEM"]]
    ] .o. LEXfilteredTemp13 ] ;
define LEXfilteredTemp15 [ ~[ $["+PRON" (?) "+3P" "+SG" "+FEM"
    ?* "+VROOT" "+SUBST" ?* "+EMPH" "+3P" "+SG"
    ["+MASC"|"+NEUT"]] ] .o. LEXfilteredTemp14 ] ;
```

C.2.21 v_all_cap_filter_imp.rule

```
#***** v_all_cap_filter_imp.rule *****
# Th. Fransen, 21/08/19
# Invoked as part of v_all.script
# Upper-level filters to select those imp. forms we expect can
    occur with a capital, i.e. independent forms not preceded by
    mutations, and only with emph. particles agreeing with the
    verb ending.
# Currently wrongly excludes independ. protot. (non-imp.)
    capitalised Timchellad etc.

regex [
$["+IMP"] .o.
~$["LEN"|"NAS"|"H"] .o.
~$["+1P" ?* "+EMPH" \"+1P"] .o.
~$["+2P" ?* "+EMPH" \"+2P"] .o.
~$["+3P" ?* "+EMPH" \"+3P"] .o.
~$["+SG" ?* "+EMPH" ? \"+SG"] .o.
~$["+PL" ?* "+EMPH" ? \"+PL"]
] ;
```

C.3 Scripts (.script)

C.3.1 added.script (regression testing)

The pre- and post-change FSTs need to be saved as old and new, respectively.

```
clear stack
regex [@"new"] - [@"old"] ;
save stack added
```

C.3.2 added_lower.script (regression testing)

Extracting changes in the lower level only. The pre- and post-change FSTs need to be saved as old and new, respectively.

```
clear stack
regex [@"new"].l - [@"old"].l ;
save stack added_lower
```

C.3.3 all.script

```
#***** all.script *****
# Th. Fransen, 21/08/19
# Combine additional lexicons (tbf) with OI verb transducer

echo >>> Opening 'all.script' ...

clear stack

# full-form lexicon with a selection of forms from Táin Bó Fraích (Meid 1974)

echo >>> Reading in 'tbf.lexc' ...
read lexc < tbf.lexc
define LEXtbf

echo >>> Capitalisation ...
define LEXtbf [ LEXtbf | [LEXtbf .o. capRule] ] ;

echo >>> Unioning 'oiv.fst' with additional lexicons
regex [@"oiv.fst" | LEXtbf] ;

echo >>> Saving stack to file ...
save stack oiAll.fst
```

C.3.4 alphabet.script

```
#***** alphabet.script *****
# Th. Fransen, 16/08/19
# Defined variables invoked in subsequent scripts and rules

echo >>> Opening 'alphabet.script' ...

clear stack

```

```
echo >>> Defining consonants and vowels ...

define nonPalCons [b|c|d|f|g|h|l|m|n|p|r|s|t|µ|θ|
"^D"|"^F"|"^M"|"^N"|"^S"] ;
define palCons [nonPalCons ' ];
define Cons [ nonPalCons | palCons ] ;
define backVowShort [a|o|u] ;
define backVowLong [á|ā|ó|ú];
define backVow [backVowShort | backVowLong ] ;
define frontVowShort [e|i] ;
define frontVowLong [é|í|ī] ;
define frontVow [frontVowShort | frontVowLong ] ;
define shortVow [ backVowShort | frontVowShort] ;
define longVow [backVowLong | frontVowLong] ;
define Vow [backVow | frontVow | ə | "^V"] ;
```

C.3.5 cap.script

```
#***** cap.script *****
# Th. Fransen, 21/08/19
# Capitalisation of initial letters

echo >>> Opening 'cap.script' ...

clear stack

echo >>> Defining capitalisation rules ...

define capRule [
# No capitalisation double non-lenited anlaut consonants
[ l -> L || .#. _ \l ] .o.
[ m -> M || .#. _ \m ] .o.
[ n -> N || .#. _ \n ] .o.
[ r -> R || .#. _ \r ] .o.

[ b -> B , c -> C , d -> D , f -> F , g -> G , h -> H , l -> L
    ,p -> P , s -> S , t -> T , a -> A , á -> Á ,  e -> E , é -> É
     , i -> I , í -> Í , o -> O , ó -> Ó , u -> U , ú -> Ú , æ ->
    Æ , ǽ -> Ǽ || .#. _  ]
] ;
```

C.3.6 `lost.script` (regression testing)

The pre- and post-change FSTs need to be saved as `old` and `new`, respectively.

```
clear stack
regex [@"old"] - [@"new"] ;
save stack lost
```

C.3.7 `lost_lower.script` (regression testing)

Extracting changes in the lower level only. The pre- and post-change FSTs need to be saved as `old` and `new`, respectively.

```
clear stack
regex [@"old"].l - [@"new"].l ;
save stack lost_lower
```

C.3.8 `mutation.script`

```
#***** mutation.script *****
# Th. Fransen, 19/08/19
# Define two-level mutation symbols (transducers) and manipulate lower-level symbols
# This script also defines non-leniting anlaut consonants

echo >>> Opening 'mutation.script' ...

clear stack

echo >>> Defining mutation transducers ...

#\\\\\ LENITION /////

# Two-level symbols
define lenTwoLevel ["+LEN":"^LEN" ] ;

# Lower-level rules
define lenLower [
[ [..] -> h || "^LEN" [c|p|t] _ ] .o.
# No lenition with sc, sp, st, sm
[ s (->) ṡ || "^LEN" _ \[c|p|t|m] ] .o.
[ f (->) ḟ|0 || "^LEN" _ ] .o.
"^LEN" -> 0
] ;

```

```
#\\\\\ NASALISATION /////

# Two-level symbols
define nasTwoLevel ["+NAS":"^NAS"] ;

# Lower-level rules
define nasLower [
[ m (->) {mm} , n (->) {nn} , r (->) {rr} , l (->) {ll} ||
"^NAS" _ ] .o.
[ [..] -> [n|ṅ] "-" || "^NAS" _ Vow ] .o.
[ [..] -> [n|ṅ]      || "^NAS" _ d|g ] .o.
[ [..] -> m || "^NAS" _ b] .o.
"^NAS" -> 0
] ;

#\\\\\ H-MUTATION /////

# Two-level symbols
define hTwoLevel ["+H":"^H"] ;

# Lower-level rules
define hLower [ [ c (->) {cc} , d (->) {dd} , f (->) {ff} , g (->) {gg} , l (->) {ll} , m (->) {mm} , n (->) {nn} , p (->) {pp} , r (->) {rr} , s (->) {ss} , t (->) {tt} || "^H" _ ] .o.
"^H" -> 0 ] ;

#\\\\\ UNION MUTATION TRANSDUCERS AND REPLACE RULES /////

define mutTwoLevel [ lenTwoLevel | nasTwoLevel | hTwoLevel ] ;

# Lower
define mutLower [ lenLower .o. nasLower .o. hLower .o. [ "^LEN"|"^NAS"|"^H" -> 0 ] ] ;

#\\\\\ NON-LEN. ANLAUT CONSONANTS /////

define nonLenAnlaut [
[ m -> {mm} , n -> {nn} , r -> {rr} , l -> {ll} || .#. _ ] ] ;
```

C.3.9 proclitic.script

```
#***** proclitic.script *****
# Th. Fransen, 16/08/19

echo >>> Opening 'proclitic.script' ...

clear stack
```

```

#\\\\\ DEFINE LEXICON /////

echo >>> Reading in 'proclitic.lexc' ...
read lexc < proclitic.lexc

echo >>> Making flag diacritics two sided ...
tfd

define LEXproclitic ;

#\\\\\ APPLY REPLACE RULES /////

set flag-is-epsilon ON

echo >>> Cleaning up initial +'s ...

# invert lexicon as replace rules target lower level, invert afterwards
define LEX [LEXproclitic.i .o. "+" -> 0 || .#. _ ].i ;

# Orthographic rules

define vowCoalesc [
["^PRONa" -> 0 || {ní} _ ] .o.
[Vow -> 0 ||  _ "^PRONa" ] .o.
["^PRONa" -> a]
] ;

define consOrth [
"^M" -> m|{mm} .o.
"^N" -> n|{nn}
] ;

define orthRules [ vowCoalesc .o. consOrth ] ;

echo >>> Applying lower-level rules ...
regex [LEX .o. orthRules] ;

set flag-is-epsilon OFF

#\\\\\ DEFINE FINAL LEXICON AS VARIABLE /////

echo >>> Defining final lexicon ...
define LEXprocl
```

C.3.10 procl_se.script

```
#***** procl_se.script *****
# Th. Fransen, 20/08/19
# Combine variables derived from proclitic.lexc (proclitic.script) and se.lexc (se_3_mut.script)

echo >>> Opening 'procl_se.script' ...

clear stack

#\\\\\ DEFINE UNFILTERED LEXICON /////

echo >>> Concatenating variables 'LEXprocl' · 'LEXseConjWithMut' ...
define LEXunfiltered [LEXprocl "+PROCL_JUNCT":· LEXseConjWithMut ] ;

#\\\\\ APPLY UPPER-LEVEL FILTER RULES /////

source procl_se_filter_1_aug.rule
define proclSEfilter1 ;

source procl_se_filter_2_no.rule
define proclSEfilter2 ;

source procl_se_filter_3_ipv.rule
define proclSEfilter3 ;

source procl_se_filter_4_pass.rule
define proclSEfilter4 ;

source procl_se_filter_5_sv.rule
define proclSEfilter5 ;

source procl_se_filter_6_mut.rule
define proclSEfilter6 ;

define proclSEfilters [proclSEfilter1 .o. proclSEfilter2 .o. proclSEfilter3 .o. proclSEfilter4 .o. proclSEfilter5 .o. proclSEfilter6 ] ;

echo >>> Filtering ...
define LEXfilteredTemp1 [proclSEfilters .o. LEXunfiltered] ;

echo >>> More filtering and defining intermediate lexicons ...
# Intermediate lexicons: LEXfilteredTemp2-15
source procl_se_filter_7_emph.rule
```

```
push LEXfilteredTemp15

define LEXproclSElegal

#\\\\\ DELETE PROCLITIC MUTATION TAGS /////

set flag-is-epsilon ON

echo >>> Deleting mirroring mutation tags before +PROCL_JUNCT ...
# invert lexicon as replace rules target lower level, invert afterwards
regex [
LEXproclSElegal.i .o.
["+NAS" -> 0 || _ "+PROCL_JUNCT" "+NAS"] .o.
["+LEN" -> 0 || _ "+PROCL_JUNCT" "+LEN"] .o.
["+H" -> 0 || _ "+PROCL_JUNCT" "+H"]
].i ;

set flag-is-epsilon OFF

#\\\\\ DEFINE FINAL LEXICON AS VARIABLE /////

echo >>> Defining final lexicon ...
define LEXproclSE
```

C.3.11 se_1_bare.script

```
#***** se_1_bare.script *****
# Th. Fransen, 16/08/19
# Initial mutations have not been added yet (hence "bare").

echo >>> Opening 'se_1_bare.script' ...

clear stack

#\\\\\ DEFINE LEXICON /////

echo >>> Reading in 'se.lexc' ...
read lexc < se.lexc

echo >>> Making flag diacritics two sided ...
tfd

echo >>> Eliminating W2a flag ...
eliminate flag W2a

define LEXunfiltered
```

```

#\\\\\ APPLY RULES /////

echo >>> Filtering ...
source 1_se_filters.rule
define RUL
regex RUL .o. LEXunfiltered ;
define LEXseLegal

set flag-is-epsilon ON

echo >>> Applying lower-level replace rules defining intermediate lexicons ...

# Phonological rules

source 2_se_phon_non_pres_stem_form.rule
define RUL2
define LEXseLower1 [LEXseLegal .o. RUL2] ;

source 3_se_phon_lowering.rule
define RUL3
define LEXseLower2 [LEXseLower1 .o. RUL3] ;

source 4_se_phon_pal.rule
define RUL4
define LEXseLower3 [LEXseLower2 .o. RUL4] ;

source 5_se_phon_del_stem_vow.rule
define RUL5
define LEXseLower4 [LEXseLower3 .o. RUL5] ;

source 6_se_phon_syncope.rule
define RUL6
define LEXseLower5 [LEXseLower4 .o. RUL6] ;

source 7_se_phon_cons_qual_assim.rule
define RUL7
define LEXseLower6 [LEXseLower5 .o. RUL7] ;

source 8_se_phon_stem_vow.rule
define RUL8
define LEXseLower7 [LEXseLower6 .o. RUL8] ;

# Orthographic rules

source 9_se_orth_vow_pal_cons.rule
define RUL9
define LEXseLower8 [LEXseLower7 .o. RUL9] ;

source 10_se_orth_end_vow.rule
define RUL10
define LEXseLower9 [LEXseLower8 .o. RUL10] ;

source 11_se_vow_emph.rule
define RUL11
define LEXseLower10 [LEXseLower9 .o. RUL11] ;

source 12_se_del_pal_markers.rule
define RUL12
define LEXseLower11 [LEXseLower10 .o. RUL12] ;

source 13_se_phon_orth_cons.rule
define RUL13
define LEXseLower12 [LEXseLower11 .o. RUL13] ;

set flag-is-epsilon OFF

echo >>> Apply priority union: robmmar > robammar ...
regex [
[
$[{ro} "+AUG" "+" {bí} "+VROOT" "+SUBST" "+PRT" "+CONJ" "+1P" "+PL"] .o. LEXseLower12 .o.
[{robmmar} -> {robammar}]
] .P. LEXseLower12
] ;

#\\\\\ DEFINE FINAL LEXICON AS VARIABLE /////

echo >>> Defining final lexicon ...
define LEXseBare
```

C.3.12 se_2_abs_conj.script

```
#***** se_2_abs_conj.script *****
# Th. Fransen, 18/08/19
# Extract absolute and conjunct endings
# Deuterotonic forms (R.PV.X) do not come up when flags obeyed

echo >>> Opening 'se_2_abs_conj.script' ...

clear stack

echo >>> Defining absolute and conjunct lexicons ...

```

```
#\\\\\ SIMPLE ABSOLUTE FORMS /////

define LEXseAbs [$["+ABS"] .o. LEXseBare] ;

#\\\\\ CONJUNCT FORMS /////

define LEXseConj [$["+CONJ"] .o. LEXseBare] ;
```

C.3.13 se_3_mut.script

```
#***** se_3_mut.script *****
# Th. Fransen, 19/08/19
# Apply mutation transducers and replace rules (mut.script) to
    absolute and conjunct forms

clear stack

echo >>> Opening 'se_3_mut.script' ...

set flag-is-epsilon ON

#\\\\\ SIMPLE ABSOLUTE RELATIVE /////

echo >>> Applying nasalisation to absolute relative forms and
    unioning lexicons ...

# Union simple absolute lexicon with nasalising absolute relative
# e.g. mbrises
define LEXabsWithNasRel [ [ LEXseAbs |
[ nasTwoLevel [$"+REL" .o. LEXseAbs] .o. nasLower ] ]
] ;

echo >>> Deleting +'s on upper level ...
# Delete remaining initial +'s on the upper level.
# invert lexicon as replace rules target lower level, invert
    afterwards
define LEXabsWithNasRelClean [ LEXabsWithNasRel.i .o.
["+" -> 0 , "+NAS" -> "NAS" || .#. _]
].i ;

#\\\\\ CONJUNCT /////

# Conj. forms (incl. imperative) are potentially separated from
    their pretonic prefix and might have received an initial
    consonant mutation.
# For strings consecutive to the 'stem', the right mutations can
    be sorted later with deleting incompatible mutation tags
```

```
(upper level).
# Most mutations are coded as optional ones, e.g. m (->) mm (nas, h-mut), except obligatory ones like t->th, d->nd etc.

echo >>> Applying mutations with conjunct forms and unioning lexicons ...
define LEXseConjWithMut  [
[ LEXseConj .o. nonLenAnlaut ] |
[ [ mutTwoLevel LEXseConj ] .o. mutLower ] |
LEXseConj
] ;

set flag-is-epsilon OFF
```

C.3.14 v_all.script

```
#***** v_all.script *****
# Th. Fransen, 21/08/19
# Create subsets of verb lexicons and load in full-form copula lexicon
# Add deuterotonic tags and perform capitalisation of initials on subsets
# Union all transducers

echo >>> Opening 'v_all.script' ...

clear stack

#\\\\\ CREATE DEUT. VS NON-DEUT. INFLECTION LEXICON AND ADD DEUT. TAGS /////

echo >>> Creating deuterotonic lexicon, adding tag ...
# add DEUT to the subset of conjunct forms that do not have the @D.PV@. flag, i.e. @R.PV.X@ flag, i.e. deuterotonic
define LEXdeutWithMut ["DEUT":0 [ ~$["@D.PV@"] .o. LEXseConjWithMut] ] ;

echo >>> Creating simple and prototonic lexicon ...
# Define a conjunct lexicon only containing simplexes and protot. forms by subtracting the deuterotonic subset (see above)
define LEXconjSimplePrototWithMut [ LEXseConjWithMut - [ ~$["@D.PV@"] .o. LEXseConjWithMut ] ] ;

#\\\\\ REWRITE UPPER STRINGS TO GET RID OF INITIAL "+" (EXCLUSIVELY WITH SIMPLEX CONJUNCT AND PROTOTONIC FORMS) /////

```

```
set flag-is-epsilon ON

echo >>>  Deleting initial +'s ...
# invert lexicon as replace rules target lower level, invert afterwards
define LEXconjSimplePrototWithMutClean [
LEXconjSimplePrototWithMut.i .o.
["+" -> 0 , "+LEN" -> "LEN" , "+NAS" -> "NAS" , "+H" -> "H" || .#. _ ]
].i ;

#\\\\\ CAPITALISATION OF INITIALS WITH RESTRICTED FORMS /////

echo >>> Defining lexicons for capitalisation ...

# Imperative

source v_all_cap_filter_imp.rule
define capFilterImpRule
define LEXimpForCap [ capFilterImpRule .o. LEXconjSimplePrototWithMutClean ] ;

# Non-relative

# Derivates of se.lexc (se_x_y.script files) or a product of proclitic.script / procl_se.script
define LEXnonRelforCap [ ~$["+REL"] .o. [LEXabsWithNasRelClean | LEXprocl | LEXproclSE] ] ;

# Copula

echo >>> Reading in 'copula.lexc' ...
read lexc < copula.lexc
define LEXcopula

# Only non-depend. copula forms (preceded by a conj. part. or aug. or not containing "+DEPEND") can be capitalised
define LEXcopulaForCap [ [ $["+CONJ_PART"|"+AUG" "+DEPEND"] | ~$"+DEPEND" ] .o. LEXcopula ] ;

echo >>> Unioning lexicons for capitalisation ...
define LEXforCap [ LEXimpForCap | LEXnonRelforCap | LEXcopulaForCap ] ;

echo >>> Capitalisation ...
define LEXcaps [LEXforCap .o. capRule] ;

set flag-is-epsilon OFF
```

```

#\\\\\ UNION TRANSDUCERS /////

echo >>> Unioning final lexicons ...
regex [
[
LEXabsWithNasRelClean |
LEXdeutWithMut.f |
# remove flag eliminator if analysis non-consecutive deut. stems not desired

LEXconjSimpleProtot WithMutClean |
LEXprocl |
LEXproclSE |
LEXcopula |
LEXcaps
]
.o. · (->) "-" .o. "-" (->) 0
] ;

echo >>> Saving stack to file ...
# Save Old Irish verbs transducer
save stack oiv.fst
```

C.4 Stem entry files (.txt)

These lists are inserted into se_empty.lexc (cf. section C.1.3) by means of a unix command as part of a shell script (cf. section C.5). The resulting lexicon is named se.lexc.

C.4.1 compoundW1.txt

```
@R.PV.AD@+ell+VROOT:ell
        W1;
@D.PV@+ad+PV1+ell+VROOT:adell
        W1;

@R.PV.TO@+ell+VROOT:all
        W1;
@D.PV@+to+PV1+ell+VROOT:tall
        W1;

@R.PV.TO@+imbi+PV2+cell+VROOT:i^Məchell
        W1;
@D.PV@+to+PV1+imbi+PV2+cell+VROOT:ti^Məchell
        W1;

```

```
@R.PV.FO@+fer+VROOT:fer
        W1;

@R.PV.IMBI@+múch+VROOT:múch
        W1;
```

C.4.2 compoundW2a.txt

```
@R.PV.ARE@+āl+VROOT:ál
        W2a;

@R.PV.ARE@+sēt+VROOT:pett
        W2a;

@R.PV.TO@+lēc+VROOT:léic
        W2a;
@R.PV.TO@+ro+AUG+lēc+VROOT:reiləc
        W2a;
@D.PV@+to+PV1+lēc+VROOT:teiləc
        W2a;
@D.PV@+to+PV1+ro+AUG+lēc+VROOT:tarələc
        W2a;

@R.PV.FO@+dāl+VROOT:dáil
        W2a;
@D.PV@+fo+PV1+dāl+VROOT:fodəl
        W2a;

@R.PV.FO@+ruim+VROOT:ruim
        W2a;

@R.PV.IMBI@+rād+VROOT:rád
        W2a;
@R.PV.IMBI@+ro+AUG+rād+VROOT:rorəd
        W2a;
```

C.4.3 simpleW1.txt

```
+an+VROOT:an                        W1;

+car+VROOT:car                      W1;

+celebrare+VROOT:celebr             W1;

+fer+VROOT:fer                      W1;
```

```

+marb+VROOT:marb                    W1;

+gat+VROOT:gat                      W1;

+íad+VROOT:íad                      W1;

+las+VROOT:las                      W1;

+marb+VROOT:marb                    W1;

+múch+VROOT:múch                    W1;

+rann+VROOT:rann                    W1;

+scar+VROOT:scar                    W1;
```

C.4.4 simpleW2a.txt

```
+aisic+VROOT:aisic                  W2a;

+bris+VROOT:bris                    W2a;

+glúais+VROOT:glúais                W2a;

+lēc+VROOT:léic                     W2a;
+ro+AUG+lēc+VROOT:reiləc            W2a;

+rād+VROOT:rád                      W2a;
+ro+AUG+rād+VROOT:rorəd             W2a;
```

C.5 Shell script and directory structure

```
#***** shell_script *****
# Th. Fransen, 23/08/19
# Script to fill se.lexc with stems and consecutively read foma scripts

echo ">>> Opening shell_script ..."

# Swap stem list files (.txt) for <PLACEHOLDERS> in se_empty.lexc, employing intermediate temp files
echo ">>> Adding stems to se.lexc ..."
```

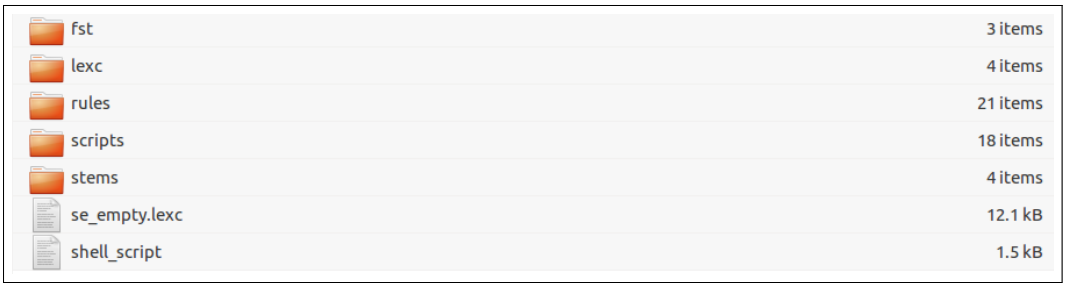
```
sed -e '/<INSERT SIMPLE W1 STEMS>/r stems/simpleW1.txt' -e '/<INSERT SIMPLE W1 STEMS>/d' se_empty.lexc > se_temp1.lexc
sed -e '/<INSERT SIMPLE W2a STEMS>/r stems/simpleW2a.txt' -e '/<INSERT SIMPLE W2a STEMS>/d' se_temp1.lexc > se_temp2.lexc
sed -e '/<INSERT COMPOUND W1 STEMS>/r stems/compoundW1.txt' -e '/<INSERT COMPOUND W1 STEMS>/d' se_temp2.lexc > se_temp3.lexc
sed -e '/<INSERT COMPOUND W2a STEMS>/r stems/compoundW2a.txt' -e '/<INSERT COMPOUND W2a STEMS>/d' se_temp3.lexc > se.lexc
mv se.lexc lexc
rm se_temp1.lexc se_temp2.lexc se_temp3.lexc

# Temporarily move lexicons and rules to scripts directory
echo ">>> Moving files to scripts directory ..."
mv lexc/*.lexc scripts
mv rules/*.rule scripts

echo ">>> Going to scripts directory ..."
cd scripts

# foma
echo ">>> Reading script files in foma ..."
foma -l alphabet.script \
foma -l mutation.script \
foma -l cap.script \
foma -l proclitic.script \
foma -l se_1_bare.script \
foma -l se_2_abs_conj.script \
foma -l se_3_mut.script \
foma -l procl_se.script \
foma -l v_all.script \
foma -l all.script \

echo ">>> Moving files back to directories ..."
cd ..
mv scripts/*.lexc lexc
mv scripts/*.rule rules
# Move saved fsts to fst directory
mv scripts/*.fst fst
```



```
fst             3 items
lexc            4 items
rules           21 items
scripts         18 items
stems           4 items
se_empty.lexc   12.1 kB
shell_script    1.5 kB
```

Figure C.1 – Directory structure for generating FSTs.