University of Brighton

ITRI-01-27

Architectures for Multilingual Lexical Representation

Carole Tiberius

September, 2001

A Doctoral Thesis

Information Technology Research Institute Technical Report Series

ITRI, Univ. of Brighton, Lewes Road, Brighton BN2 4GJ, UK
TEL: +44 1273 642900 EMAIL: firstname.lastname@itri.brighton.ac.uk
FAX: +44 1273 642908 NET: http://www.itri.brighton.ac.uk

Architectures for Multilingual Lexical Representation

Carola Petra Adriënne Tiberius

A thesis submitted in partial fulfilment of the requirements of the University of Brighton for the degree of Doctor of Philosophy

October 2001

Information Technology Research Institute
University of Brighton

Abstract

This thesis is an investigation of the use of inheritance networks to construct a multilingual lexicon in which information can be shared at several levels of linguistic description – morphology, phonology, syntax, etc. It aims to provide an insight into methodological and theoretical issues involved in the development of such lexicons, in particular:

- The regulation of the inter- and intralanguage inheritance relations
  That is, how does the hierarchical structure of one language interact with the hierarchical structure between the languages?

- Multilingual information sharing
  Which information can and should be shared in a multilingual inheritance network and how is it shared?

- Development strategies
  How does one go about constructing a multilingual inheritance lexicon? Should the monolingual and multilingual hierarchical lexicons be developed in parallel and linked immediately upon construction or should a non-parallel development strategy be adopted, where the monolingual lexicons are first fully developed separately and only linked together at the end.

This thesis explores these issues by comparing different architectures for multilingual inheritance lexicons following Evans' (1996) proposals. A parameterised approach, in which language is used as a parameter, is contrasted with a non-parameterised approach by implementing sample fragments in DATR (Evans and Gazdar, 1996). The sample fragments focus on the sharing of morphological, phonological, and morphophonological similarities and cover a small set of nouns and adjectives in Dutch, English, Danish, and Icelandic.

Contents

Abstract ii

Contents iii

List of Figures vii

List of Tables x

Acknowledgements xi

Author's declaration xii

1 Introduction 1

1.1 Aims 1

1.2 Motivation 4

1.3 Overview of the thesis 6

2 Multilingual Lexicons 9

2.1 Introduction 9

2.2 Standardisation 9

2.3 Multilingual Lexicons for MT 14

2.3.1 The direct model 16

2.3.2 The transfer model 16

2.3.3 The interlingua model 17
2.4 Inheritance-Based Approaches to Multilingual Lexicons 19
2.4.1 Kameyama's multilingual unification grammar for nominal expressions 21
2.4.2 PolyLex 23
2.4.3 GREG 25
2.5 Other related work 25
2.6 Summary 27

3 Multilingual Architectures **28**
3.1 Introduction 28
3.2 Inheritance-based formalisms 28
3.3 Inheritance in DATR 32
3.4 Multilingual Inheritance 33
3.4.1 Non-parameterised multilingual inheritance 37
3.4.2 Parameterised multilingual inheritance 39
3.5 Conclusion 46

4 Methodology **48**
4.1 Introduction 48
4.2 Data Selection 48
4.2.1 Language Sampling 49
4.2.2 Lexical Sampling 56
4.3 Multilingual Information Sharing 59
4.4 Development Strategy 62
4.5 Evaluation 63
4.6 Summary 64

5 The Lexical Description Framework **65**

5.1 Introduction 65

5.2 The lexical description framework 66

5.2.1 Theoretical Background 66

5.2.2 Objects of description 68

5.2.3 The organisational structure of the lexicon 69

5.3 Implementation of the framework in DATR 72

5.4 Illustration of the framework 75

5.5 Summary 90

6 Implementation and Evaluation **91**

6.1 Introduction 91

6.2 The implementation in general 91

6.3 Implementation of the multilingual architectures 96

6.3.1 The Structure-Sharing Model 96

6.3.2 General comments on the implementation of the parameterised models 98

6.3.3 The Meta-Features Model 101

6.3.4 The Infinitesimal model: A restricted version 103

6.3.5 The MetaTheory Model 106

6.4 Evaluation 108

6.5 Conclusion 113

7 Concluding remarks **116**

7.1 Contribution of the thesis 116

7.1.1 Regulation of the inter- and intralanguage inheritance 117

7.1.2 Multilingual Information Sharing 120

7.1.3 Development Strategy 121
7.1.4 Further contributions 122
7.2 Future development 122
7.2.1 Further exploration of the models 123
7.2.2 Multilingual Information Sharing 123
7.2.3 The status of the syllable in multilingual lexical representation 124
7.2.4 Multilingual lexical representation and semantics 126
7.2.5 Machine Learning and Multilingual Lexical Representation . 127
7.2.6 Towards a multilingual featural description 127
7.3 Summary and Conclusions 128

A General Implementation Conventions **130**

B Test Data **133**

B.1 Body Parts Test Set 134
B.2 Medical Test Set 137

Bibliography **152**

List of Figures

1.1 Multilingual architecture with conditionalised partitions 3

1.2 Schematic representation of a non-parameterised model 3

2.1 The MULTILEX multilingual linguistic architecture (Ahmad et al., 1993, p.11) 11

2.2 General architecture of a PAROLE lexicon (Ruimy et al., 1998, p.242) 14

2.3 Template associated with the concept *instrument* (Bel et al., 2000, p. 1381) 15

2.4 MT pyramid 15

2.5 EuroWordNet architecture (Vossen 1998, p.80) 19

2.6 Shared Grammar from Kameyama (1988, p.195) 22

2.7 Multilingual Inheritance Hierarchy (Cahill and Gazdar, 1999b, p.14) 24

3.1 Monotonic Single Inheritance Network 29

3.2 Non-Monotonic Single Inheritance Network 29

3.3 Monotonic Multiple Inheritance Network 30

3.4 Non-Monotonic Multiple Inheritance Network 30

3.5 A language typology 34

3.6 Non-parameterised multilingual inheritance hierarchy with a flat language typology 35

3.7 Non-parameterised multilingual inheritance hierarchy with subhierarchies 35

3.8 Parameterised multilingual inheritance hierarchy 36
3.9 Classification of the Germanic languages 39
3.10 Illustration of the Micro-Features model 40
3.11 Feature tree for verb morphology in Dutch, English, and German . 41
3.12 Feature tree for nouns in German 41
3.13 Feature tree for nouns in English and Dutch 42
3.14 Illustration of the Meta-Features model 42
3.15 Feature tree for nouns in English and Dutch 43
3.16 Feature tree for nouns in German 43
3.17 Illustration of the Infinitesimal model 44
3.18 Parameterised tree structure for a subset of the Germanic languages 45
3.19 Example of the MetaTheory model 46

4.1 Tree transformation 50
4.2 Width of the three intermediate levels of the BC subfamily 51
4.3 Classification of the Germanic language family according to the Ethnologue (1996) 54
4.4 Width of the four intermediate levels of the Germanic language family 55
4.5 Lexical entry for *Cat* in a data-driven approach 62

5.1 Conventional syllable structure 68
5.2 Wordform structure for *Hand* 69
5.3 Word form structure for *fingers* 70
5.4 Module and node structure for lexeme *Gebed* 71
5.5 Example of rule chaining 74
5.6 Fragment of a lexical hierarchy of Dutch noun inflection 76
5.7 Final devoicing applied to the Dutch lexeme *Gebed* ('prayer') 82

5.8 Plural and definite rules applied to the Danish lexeme *Mund* /mOn/ ('mouth') 86

5.9 Definition of `plural_s` rule 88

6.1 Lexeme hierarchy in the SS/BODYPART lexicon 97

6.2 Adding a dialect to a Structure-Sharing model 97

6.3 Language tree used in the implementation 100

6.4 Noun Hierarchy 1 100

6.5 Noun Hierarchy 2 100

6.6 Feature space in the Meta-Features Model 101

6.7 Feature space in the Infinitesimal Model 103

6.8 Extract of the inheritance hierarchy of the MetaTheory fragment . . 107

6.9 Revised language typology for our test sets 111

7.1 Extract of the actual noun class hierarchy in the SS/BODYPART implementation 118

7.2 Extract of the actual noun class hierarchy in the META/BODYPART implementation 118

7.3 Multiple inheritance network with semantics 126

A.1 Lexical Description Framework with illustration of node naming conventions 131

List of Tables

3.1 Lexeme entries for *Bishop* and *Fish* (Cahill and Gazdar, 1999b, p.18). 38

4.1 Computation of DV of the West Germanic language family 53

4.2 Computation of DV of the North Germanic language family 53

6.1 Phonemic transcriptions for the lexical entry *Diameter* 93

6.2 Phonemic transcriptions for the lexical entry *Perforation* 94

6.3 Phonemic transcription for the lexical entry *Elbow* 95

6.4 Phonemic transcriptions for the lexical entry *Contact* 95

6.5 Overview of the number of statements per node 109

6.6 Overview of the number of statements per lexical node 109

6.7 Overview of inferential complexity 110

7.1 Information sharing based on SS/MED 121

7.2 Phonemic transcriptions for the lexical entry *Diameter* 124

Acknowledgements

There are many people to whom I owe my gratitude. First, I would like to thank my supervisors Roger Evans, Gerald Gazdar and Adam Kilgarriff for the valuable feedback they have provided throughout my doctorate. I have learned a lot from them. I extend my thanks to Lynne Cahill for her interest in my thesis and her support throughout. I also wish to thank my examiners, Julie Carson-Berndsen and Bill Keller, for their comments.

Thanks are also due to the University of Brighton and the EPSRC for their financial support without which this research would not have been possible.

I would like to thank Robert Vander Stichele for providing me with the Dutch, English, and Danish files of the medical glossary that I used as the basis for one of my test sets, Peter Molbæk Hansen for allowing me to use a machine-readable copy of his Danish pronunciation dictionary, Inger Lytje for checking the Danish data and Margrét Jónsdóttir for checking the Icelandic data.

I am also extremely grateful to Greville Corbett and Dunstan Brown from the University of Surrey for helping me complete this thesis.

Last, but certainly not least, special thanks go to my family and friends.

Author's declaration

This thesis is the result of my own work, except for the material stated below which is based on work done in collaboration.

The lexical description framework discussed in this thesis is based on joint work with Roger Evans and draws on material presented as Tiberius and Evans (2000). Roger Evans was responsible for the development of the abstract framework and the implementation of the framework in DATR. The author of this thesis was responsible for the linguistic design, including aspects such as data collection, implementation of lexical sample fragments, testing and evaluation of the framework.

This thesis further explores the concept of metaphonemes as introduced in Tiberius and Cahill (2000a; 2000b). The concept of metaphonemes originates from earlier work by Lynne Cahill on the PolyLex project (Cahill and Gazdar, 1999b). In the context of this thesis, the author defined cross-linguistic phoneme correspondences for the vowel phonemes in Dutch, English, and Danish, and incorporated these in the lexical sample fragments.

Note that appendix C – containing a copy of Tiberius and Evans (2000) – and D – containing the DATR sample fragments – of the original thesis are not included in this version. A copy of either can be obtained from the author.

Chapter 1

Introduction

1.1 Aims

It is now well-established in work on theoretical lexical representation that it is useful to see a lexicon as some kind of inheritance network (see for example, Daelemans and Gazdar (1992), Briscoe et al. (1993)). The syntax, semantics, and morphology of most words is shared with that of many others and it is necessary for a theoretically adequate lexical representation language to be able to capture these generalisations. Inheritance networks allows us to do so. The rationale of inheritance-based lexicons requires that information is pushed as far up the hierarchy as it can go capturing as many generalisations as possible. In a monolingual lexicon, this means that information which applies to all words of a language will appear right at the top of the hierarchy, information that is common to all nouns appears above all the individual noun entries and so on. The same rationale can be applied in a multilingual context to capture cross-linguistic similarities – information which is common to all languages in the lexicon will be stated at higher points in the hierarchy than that which is unique to just one of the languages.

Research into the application of inheritance networks to multilingual lexical description is relatively new and many theoretical and methodological issues are still unanswered. This thesis focuses on three of those issues:

- **The regulation of the inter- and intralanguage inheritance relations**
  In a multilingual inheritance network, there are not just inheritance relations within a language, but there are also inheritance relations between languages.

The question is how are these two kinds of inheritance relations expressed and how do they interact? In other words, how does the hierarchical structure of one language interact with the hierarchical structure between the languages?

- **Multilingual information sharing**
  This issue can be subdivided into three questions. First, which information can be shared in a multilingual inheritance lexicon and can different architectures share the same information? In theory, a multilingual inheritance network can share information at all levels of linguistic description – syntax, semantics, morphology, phonology, etc. Second, which information should be shared in a multilingual inheritance hierarchy? Languages exhibit similarities for several reasons – genetics, typology, language contact or chance. Should a multilingual lexicon capture all these similarities or should only a specific kind of similarity be captured? Third, how is the information shared in a multilingual inheritance network? That is, which criteria are used to decide when information should be stated in a shared part?

- **Development strategies**
  How does one go about constructing a multilingual inheritance lexicon? Should the monolingual and multilingual hierarchical lexicons be developed in parallel and linked immediately upon construction or should a non-parallel development strategy be adopted, where the monolingual lexicons are first fully developed separately and only linked together at the end?

These issues are explored by comparing different architectures for multilingual inheritance lexicons following Evans' (1996) proposals.

Evans distinguishes what he calls **parameterised** and **non-parameterised** architectures for multilingual inheritance-based lexicons. In a parameterised model, linguistic descriptions are made multilingual by conditionalising the linguistic objects for language. The information that is valid for one particular language is then the collection of specifications that mention that language. Figure 1.1 shows a schematic representation of a parameterised model. This figure represents that in English nouns are only inflected for number, whereas in German, they also inflect for case. Linguistic objects can be conditionalised for language in different ways.

In a non-parameterised model, on the other hand, language is not explicitly used as a parameter. This is schematically represented in Figure 1.2. The language-specific parts are separate hierarchies which are linked together through a common

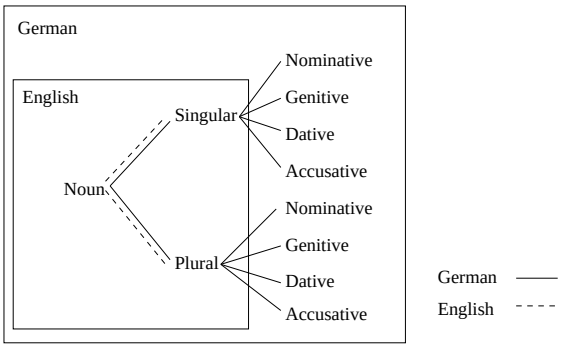


Figure 1.1: Multilingual architecture with conditionalised partitions

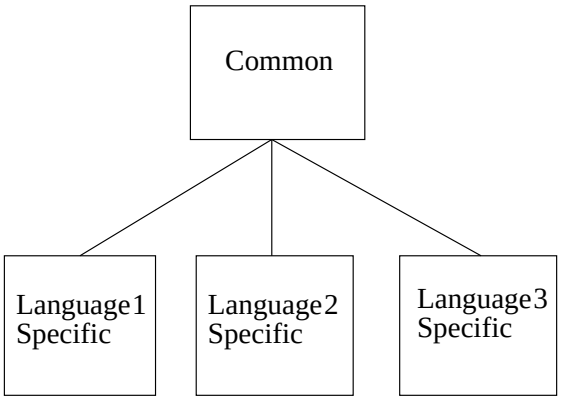


Figure 1.2: Schematic representation of a non-parameterised model

hierarchy which contains what the language-specific hierarchies have in common. There is nothing in this model that ties linguistic objects explicitly to a particular language. You have to know which hierarchy you are in to know which language you are talking about. The parameterised and non-parameterised approach to multilingual lexical representation will be further discussed in Chapter 3.

In order to compare the different architectures and to find out what their advantages and disadvantages are, sample lexical fragments have been implemented. The sample fragments are implemented in DATR, an inheritance-based formalism developed by Evans and Gazdar (1996). Two data sets were used. A small data set consisting of 19 body part terms (all nouns) in Dutch, Danish, English, and Icelandic, and a larger data set of 100 medical terms (nouns and adjectives) in Dutch, Danish, and English. The sample fragments focus on the sharing of morphological, phonological, and morphophonological information between these languages.

1.2 Motivation

Lexical research has become more and more important in language engineering. The reasons for this trend are both theoretical and practical (Briscoe, 1991). Current NLP work no longer considers the lexicon as a simple word list (in contrast to linguistic work in the 1960s and 1970s), but places it at the centre of attention, assuming that almost all of the morphology, syntax, semantics, and phonology of a language is to be captured within the lexicon rather than in extralexical components. From a practical point of view, lexicons have become bigger and bigger in order to provide an adequate vocabulary for the ever growing NLP systems. In addition, our increasingly multilingual society (with multi-national companies and organisations such as the European Union) requires information to be available in a diversity of languages and hence multilingual facilities for creating, exchanging, and accessing information across language borders are needed.

These considerations resulted in the development of Lexical Knowledge Representation Languages (LKRLs). Inspired by work on semantic nets in Artificial Intelligence, LKRLs use representation techniques involving notions of inheritance and default to capture linguistic generalisations.

So far most of the work on the application of inheritance networks to multilingual

lexical description has concentrated on sense linkage between essentially monolingual lexicons (Copestake et al., 1992), similar to the work that has been done on multilingual lexicon development for practical applications such as MT and multilingual Natural Language Generation. Very little attention has been paid to the use of inheritance networks to share information between languages at levels of linguistic description other than semantics. However, it is well-known that languages also possess similarities in their syntax, morphology, phonology, etc. Capturing such similarities can help to create more robust, more readily maintainable and more readily extensible NLP systems for natural languages (Cahill and Gazdar, 1999b).

This idea was first explored in Kameyama (1988). She describes a multilingual unification grammar for nominal expressions in Arabic, English, French, German, and Japanese. Kameyama's grammar is an example of a non-parameterised model. The multilingual grammar consists of a complex lattice of shared grammar templates expressing shared primitives and each language inherits the relevant parts to form language-specific templates. Typologically non-related languages were chosen to demonstrate the feasibility of a shared grammar for dissimilar languages.

Cahill and Gazdar (1999b) follow a similar line of research, but focus on related languages. They propose an architecture which allows the encoding and exploitation of lexical similarities between closely related languages by using an inheritance-based framework. In broad terms, a multilingual lexicon is constructed by taking a set of monolingual hierarchical lexicons and creating a parallel hierarchy which contains what the monolingual lexicons have in common. Their architecture has been applied in the PolyLex project[1] which developed a trilingual hierarchical lexicon for Dutch, English, and German sharing morphological, phonological and morphophonological information between these languages. Like Kameyama's grammar, the PolyLex architecture is an example of a non-parameterised model as it does not explicitly use language as a feature.

More recently, a similar non-parameterised multilingual architecture has been adopted in the GREG project (Kilgarriff, Cahill, and Evans, 1999). The GREG project developed a multilingual valency lexicon for Georgian, Russian, English, and German in which valency information is shared between lexical entries within and across the four languages.

[1] `http://www.cogs.susx.ac.uk/lab/nlp/polylex`.

These three projects form the main background to our research and will be discussed in more detail in Chapter 2. In this thesis, we contrast a non-parameterised approach to multilingual lexical representation with a parameterised one and determine what their advantages and disadvantages are. This way, we aim to provide an insight in the theoretical and methodological issues raised above.

1.3 Overview of the thesis

The thesis is organised as follows.

Chapter 2 – Literature Review

This chapter discusses a variety of previous work on multilingual lexicons and shows how the work described in this thesis fits into the field in general. It starts off with a short summary of efforts in standardisation and reusability, describing multilingual European projects such as GENELEX, MULTILEX and EAGLES. Then multilingual lexicons as used in MT are described, followed by a detailed discussion of Kameyama's grammar, the PolyLex project, and the GREG project, which form the main background to this thesis. The chapter concludes with a few examples of related work outside the area of lexicons.

Chapter 3 – Architectures

In this chapter, the basic architectures for multilingual lexical representation that are studied in this thesis are defined. This chapter discusses and refines the proposals of Evans (1996). We follow Evans' distinction into parameterised and non-parameterised approaches to multilingual inheritance lexicons. We consider one non-parameterised model, the Structure-Sharing model, and four parameterised models, the Micro-Features model, the Meta-Features model, the Infinitesimal model, and the MetaTheory model. A definition of each model is provided in Chapter 3.

Chapter 4 – Methodology

In order to compare the different architectures, sample lexical fragments have been implemented. Chapter 4 discusses the methodological issues involved in the implementation of the sample lexicons. First, the selection of the test data is described. To make sure that the resulting framework is generally applicable it is important to avoid any bias towards a particular language or sublanguage in the sample lexicons. The language sampling and lexical sampling strategies that have been used are discussed here. Then, the chapter looks into the problem of information sharing in a multilingual inheritance-based lexicon and the approach adopted in this thesis is described. This is followed by a discussion of development strategies and the chapter concludes with establishing measures for the evaluation of the implemented lexical fragments.

Chapter 5 – Lexical Description Framework

Chapter 5 gives a detailed description of the lexical description framework that we use in the sample fragments. It can be seen as a development of the framework used in the PolyLex project, extending PolyLex's word model down to the level of phonological features and adopting a more modular and more uniform phonologically-based approach to lexical generalisation. Like PolyLex we focus in our sample lexicons on the sharing of morphological, phonological, and morphophonological similarities. In order to capture as many similarities as possible, at as many levels (e.g. syllable, phoneme, phonological feature) as possible, an extremely flexible mechanism is required to refer to the different parts of information that make up a lexical entry.

The lexical description framework that we use organises the lexicon into distinct self-contained modules corresponding to levels of lexical description (lexemes, syllable sequences, syllables, and phonemes). Each module forms its own independent inheritance hierarchy such that generalisations can be captured at each level. Higher level relationships between word forms are represented by means of lexical rules. This way, the framework provides a flexible means of capturing lexical generalisations within and across languages. The framework has been implemented in DATR.

Chapter 6 – Implementation and Evaluation

Sample fragments of the different architectures discussed in Chapter 3 have been implemented in DATR. In Chapter 6 we look at issues involved in the implementation and evaluate the sample fragments using the metrics described in Chapter 4.

Chapter 7 – Conclusion and Future Directions

This final chapter summarises the contributions made by this thesis. Multilingual lexical representation being a fairly new area of research, this thesis can be no more than an interim exploration, and of course a lot remains to be done. The chapter concludes by outlining some of the directions in which the work might be extended.

Chapter 2

Multilingual Lexicons

2.1 Introduction

This chapter provides relevant background to the approach to multilingual lexical representation adopted in this thesis. We start our overview with a brief summary of efforts in reusability and standardisation. The underlying idea is that standards would facilitate exchange, sharing, and reusability of linguistic data. Then we move on to multilingual lexicons as used in Machine Translation (MT). Depending on the particular translation strategy that is used in an MT system, different architectures are used to structure the information in the multilingual lexicons. MT architectures for multilingual lexicons will be discussed in Section 2.3. Most lexicons that are built for MT are linked at the level of semantics only. However, as we mentioned earlier, languages may also exhibit similarities at other levels of linguistic description such as phonology, morphology, and syntax, and all these similarities could be captured in a multilingual inheritance network. This idea has been explored in Kameyama's (1988) multilingual unification grammar, in the PolyLex project (Cahill and Gazdar, 1999b), and in the GREG project (Kilgarriff, Cahill, and Evans, 1999). These projects will be discussed next. We conclude this chapter with related work outside the area of lexicons.

2.2 Standardisation

At the end of the 1980s, **reusability** of lexical resources had become a buzzword and standardisation was seen as a way to achieve this. International standards in

dictionary data representation would facilitate exchange, sharing, and reusability of lexical data. A uniform dictionary representation format would also allow dictionaries conforming to this format to be linked intelligently together in a multilingual framework. The political goals of the European Union, requiring multilingual facilities for creating, exchanging and accessing information across language borders, made it one of the main advocates of standardisation and reusability. It initiated projects such as EUROTRA, MULTILEX, GENELEX, and more recently EAGLES[1], PAROLE and SIMPLE[2]. A brief summary of each of these projects will be given below.

EUROTRA

The aim of the EUROTRA (Allegranza, Krauwer, and Steiner, 1991) project (1982 - 1990) was to develop an MT system which translated between all the languages of the European Union. Although EUROTRA was not in the first place about standardisation and reusability, it is mentioned here because of its contribution to the promotion of computational linguistics and in particular contrastive linguistics throughout the member states of the European Union. Within EUROTRA a lot of effort was spent on fundamental linguistic research in all the languages of the European Union at the time. Up to then, research had mainly focussed on the development of resources for English.

MULTILEX and GENELEX

MULTILEX (Ahmad et al., 1993) and GENELEX (Antoni-Lay, Francopoulo, and Zaysser, 1994) were two big multilingual dictionary projects which aimed to set up standards for the description of lexical entries in dictionaries for a number of European languages. A large part of the MULTILEX and GENELEX efforts was put into closely examining the morphosyntactic characteristics of a set of European languages and establishing which features and values were relevant for each language and should be used in the morphosyntactic description of that language. MULTILEX and GENELEX are very similar in spirit and an overview of their similarities and differences can be found in Menon and Modiano (1993). Here we will give a short description of the MULTILEX project.

In MULTILEX lexical information is organised around two nodes, one containing morphological, phonological, and orthographic information, called GPMU

[1]Papers about the EAGLES project can be found on the EAGLES website `http://www.ilc.pi.cnr.it/EAGLES/home.html`.

[2]More information about the SIMPLE project can be found on the SIMPLE website `http://www.ub.es/gilcub/SIMPLE/simple2.html`.

(Graphic, Phonological, Morphological Unit) and one containing syntactic and semantic information, called LU (Lexical Unit). This architecture is used in all monolingual lexicons complying with the MULTILEX standards. A multilingual resource is created by establishing translation equivalents between the monolingual lexicons. This situation is illustrated in Figure 2.1. Here we have a multilingual

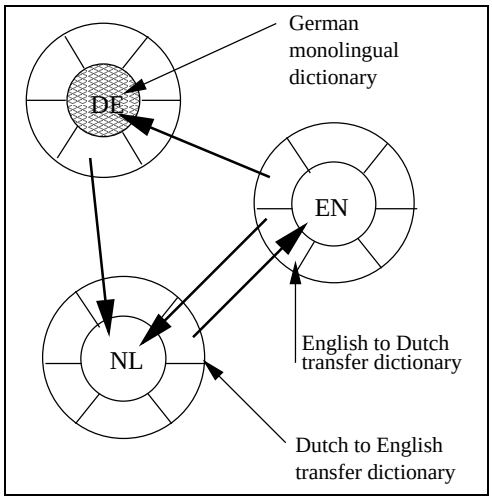


Figure 2.1: The MULTILEX multilingual linguistic architecture (Ahmad et al., 1993, p.11)

lexicon for Dutch, English, and German which consists of three monolingual lexicons plus a set of transfer dictionaries, i.e. dictionaries which define translation equivalents between two languages. The monolingual lexicons form the inner circles. The German monolingual lexicon is highlighted. The outer ring of each circle represents the transfer dictionaries. Since the transfer dictionaries are unidirectional, two transfer dictionaries are needed per language pair. For example, in the picture, we see a transfer dictionary from Dutch to English and one from English to Dutch. The kind of information that is shared in such a multilingual lexicon is restricted to semantics and a set of morphosyntactic features.

EAGLES

The EAGLES (Expert Advisory Group on Language Engineering Standards) initiative was launched in 1993 and aimed to accelerate the provision of standards for several areas of natural language engineering, including text corpora, grammar formalisms, evaluation, and spoken language resources, and computational lexicons. Each of those areas was covered by a special Working Group.

Like the MULTILEX and GENELEX projects, the original EAGLES Lexicon Working Group focussed on the morphosyntactic description of lexical items and their standardisation. They compared the main encoding practices for morphosyntactic information in lexicons and corpora and defined a consensual proposal on the basis of this comparison. This proposal was then tested by applying it to a set of European languages (including Catalan, Dutch, Greek, and Portuguese).

To be able to account for language-specific characteristics and for the different levels of granularity relevant for different languages at different levels of morphosyntactic description, a four-layered system of recommendation types was proposed. Going from language-neutral to language-specific, the following levels are distinguished. The first level (`L0`) describes **obligatory** morphosyntactic specifications. The only information which is considered obligatory for all languages is category information (e.g. noun, verb). The next level (`L1`) contains **recommended** features. For nouns this includes agreement features such as `gender` and `number`. The third level expresses **optional** information and is divided into two sublevels (`L2a` and `L2b`). `L2a` contains information which is pertinent to all or many languages and is considered useful and easy to standardise (e.g. `case`). `L2b` contains language-specific features such as `inflection type`. This is used, for example, to characterise Danish and German nouns as `weak`, `strong` or `mixed`.

Despite the relatively flexible four-layered encoding scheme, there is still room for different interpretations within the proposed 'standard'. For example, there is generally agreement on the category of the major classes (verb, noun) across languages and lexicons, but this is not the case with more minor word classes such as determiner and articles which could be collapsed into one category in some languages. Consequently, the EAGLES proposal is essentially a list of morphosyntactic features and their values that should possibly be included in the lexicon. Also, since the proposal is only based on a set of Indo-European languages found in Europe, its applicability to other languages and language families is probably limited. A complete overview of the EAGLES guidelines on morphosyntax can be found in (Eagles, 1996b). The standardisation of subcategorisation information in the lexicon is discussed in (Eagles, 1996a).

In 1999, the final report of the EAGLES Lexicon Working group on the standardisation of lexical semantic encoding for Human Language Technology applications was published. The report (Eagles, 1999) provides a survey of the issues at stake in the standardisation of lexical semantics plus an overview of a variety of resources

that contain lexical semantic information, all with specific reference to MT and Information Systems. At the end of the report, an attempt is made to define common protocols for the encoding of lexical semantic information.

The EAGLES guidelines say very little about lexical representation and architecture. A separate report was produced on Lexicon Architectures (Menon and Modiano, 1993), but this report is basically a review of the similarities and differences between MULTILEX and GENELEX. Apart from emphasising that the architecture should be modular, and language- and theory-independent, no specific guidelines were provided. Nor does EAGLES prescribe the use of classificatory devices, such as inheritance, although it recognises their importance for capturing linguistic generalisations. Work on lexical representation was left to the Working Group on Computational Linguistic Formalisms. Their guidelines, however, are mostly grammar-oriented. They extensively reviewed existing theories and systems, identified trends towards convergence, and proposed various exchange formats so that there are standard ways of representing a given theory. For example, they define *The EAGLES Encoding Format for HPSG* which includes encoding specifications for lexical entries as well as other elements of the grammar.

That the EAGLES guidelines say little about lexical representation and architectures is not so surprising considering EAGLES' goals. EAGLES' goals are primarily practical and the EAGLES Lexicon Working Group is mainly concerned with establishing standards of what should be encoded in a lexicon and how it should be encoded to support reuse, not how it should be implemented. All specifications are encoded in an SGML formal grammar which names permissible elements and describes permissible relations between them. The specific implementation of a lexicon is left to the users. The focus of this thesis is multilingual lexical representation. We are committed to an inheritance-based formalism and explore how such a formalism can be used in a multilingual context. Our aim is not to develop a lexicon for a practical NLP system as such.

The EAGLES guidelines have been implemented in the PAROLE (Calzolari, 1998) and SIMPLE (Bel et al., 2000) projects. The PAROLE project produced corpora for 14 languages and lexicons for 12 languages of the European Union using the same design principles, linguistic specifications and representation format for all the languages. Each of the 12 monolingual lexicons consists of 20,000 entries providing morphological, syntactic, and, in a few cases, semantic information. The general architecture of the PAROLE lexicons is depicted in Figure 2.2.

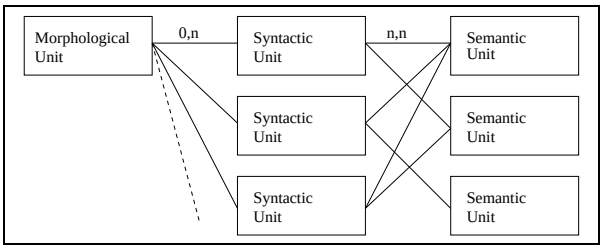


Figure 2.2: General architecture of a PAROLE lexicon (Ruimy et al., 1998, p.242)

Each lexicon consists of three independent modules (a Morphological Unit, a Syntactic Unit, and a Semantic Unit) which are linked together. A morphological unit is linked to one or more syntactic units which share the same morphological information. A syntactic unit has access to its morphological information through the link to the morphological unit it is associated with. A syntactic unit is also associated with one or more semantic units, depending on the number of meanings which can be distinguished for a single syntactic structure of a lemma. Each semantic unit, in its turn, has access to the syntactic information of the entry it is linked with. The amount of semantic information in the PAROLE lexicons is limited as this was not the main goal of the project. A semantic layer was added in the SIMPLE project, a follow-up to the PAROLE project. The lexicons in the SIMPLE project all share the same core ontology which is based on the results of the EuroWordNet project (Vossen, 1998). EuroWordNet will be discussed briefly below in Section 2.3.3. In addition, the SIMPLE project has created a common library of language-independent templates. These templates are schematic structures which reflect the constraints and conditions of well-formedness of lexical items. They are intended to facilitate the lexicographic work. An example of such a template is given in Figure 2.3.

2.3 Multilingual Lexicons for MT

Most work on multilingual lexicons has been undertaken for practical NLP applications such as MT. The multilingual lexicons constructed in this context, have mainly focussed on establishing translation equivalents between languages and they might therefore be better described as sets of semantically linked monolingual lexicons.

Usem:	1
Template_Type: **Unification_path:**	[**Instrument**] [**Concrete_entity** — **Artifact**_*Agentive* — **Telic**]
Domain:	*General*
Semantic Class:	<Nil>
Gloss:	*//free//*
Pred_Rep:	<Nil>
Selectional_Restr.:	<Nil>
Derivation:	<Nil>
Formal:	*isa* (1, <instrument>)
Agentive:	*created_by*(1, <Usem>:[Creation])

Constitutive:	*made_of*(1,<Usem>) //optional//
	has_as_part(1, <Usem>)
	//optional//

Telic:	*used_for*(1,<Usem>:[Event])
Synonymy:	<Nil>
Collocates:	*Collocates*(<Usem1>,...,<Usemn>)
Complex:	<Nil> //for regular polysemy//

Figure 2.3: Template associated with the concept *instrument* (Bel et al., 2000, p. 1381)

There are broadly three strategies that are used in MT to map the semantics of one language onto that of another – one direct method, known as the **direct** or **transformer** model, and two indirect methods, the **transfer** model and the **interlingua** model. The pyramid diagram in Figure 2.4 is often used to illustrate these different strategies.

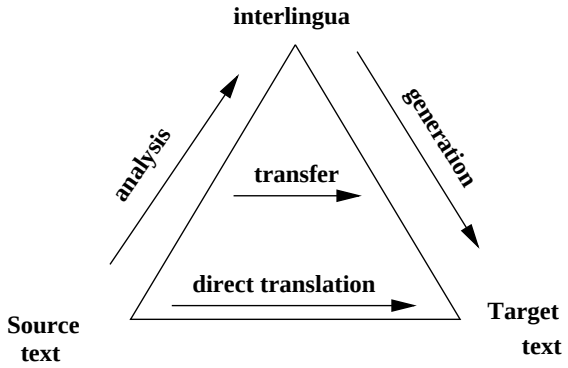


Figure 2.4: MT pyramid

The different strategies have implications for the organisation of lexical data in an MT system. We will discuss each strategy briefly below, focussing on the kind of lexicons that they require.

2.3.1 The direct model

The direct approach is historically the earliest. In this approach, there are no intermediate stages in the translation process: the processing of the source language text leads 'directly' to the desired target language output, by replacing source words with their target language equivalents followed by some local word-order adjustment. The main components of direct systems are large bilingual dictionaries containing lexical equivalences between source and target language. Often these dictionaries are contextual dictionaries which provide grammatical and collocational data to enable analyses or translations to be modified according to the context. The result can be a lexicon of considerable complexity. As noted in Arnold et al. (1994), the direct method is really designed for translation in one direction taking advantage of the similarities in structure and vocabulary between source and target language.

2.3.2 The transfer model

The first variant of the indirect approach that we discuss is called the **transfer** model. Systems which use this model, interpose bilingual transfer modules between language-dependent intermediate representations. The result of the analysis of the source text is an abstract representation which is the input to the bilingual transfer module. The transfer module then converts this source language representation into an abstract target language representation which is used to generate the target language output. This means that the system contains independent analysis and generation modules, separate monolingual lexicons for source and target language plus a bilingual transfer lexicon. The monolingual lexicons contain information necessary for, respectively, analysis and generation, while the bilingual transfer lexicon contains translation equivalents and some grammatical information for the two languages. Depending on the amount of work that is done in analysis and generation, the grammatical information in the transfer phase will be more or less complex.

The disadvantage of the transfer method is that when new languages are added to the system, not only do new modules have to be developed for the analysis and generation of those languages, but also new transfer modules have to be added establishing translation equivalents for each language pair.

The transfer model has been used in many MT systems, e.g. SUSY (Maas, 1987), METEO (Chandioux, 1976; Chevalier, Dansereau, and Poulin, 1978), EUROTRA, METAL (Bennett and Slocum, 1985; Slocum, 1987). Many multilingual dictionary projects, are also compatible with the transfer model e.g. MULTILEX GENELEX and ACQUILEX (Copestake et al., 1992). ACQUILEX defines a method for constructing a multilingual lexical knowledge base (LKB) semi-automatically from various machine-readable dictionaries. All information in the LKB is expressed in a common Lexical Representation Language with a common type system, which makes it possible to compare lexical entries within a language and to establish multilingual translation links to create an integrated multilingual LKB. ACQUILEX constructs a transfer lexicon, but it abstracts away from the traditional MT transfer rules in the sense that it allows inheritance between classes of lexical entries and makes it possible to express cross-linguistic generalisations between lexical processes.

2.3.3 The interlingua model

The second variant of the indirect method is the **interlingua** model. Rather than linking the languages pair-wise, the source text is analysed into a language-independent representation from which the target text is directly generated. The method is interlingual in the sense that the intermediate representation is neutral between two or more languages.

At the lexical level, several intermediate representations can be used (Vossen, Díez-Orzas, and Peters, 1997). First, the languages can be linked through a language-independent conceptual system or ontology. This approach has been adopted in the Mikrokosmos MT project (Mahesh, 1996; Nirenburg et al., 1996). The advantage of using an ontology is that adding new languages is relatively easy. Only two new modules need to be added per language, an analysis and a generation module. There is no need for transfer components between the new language and the languages already available in the system. The difficulty with using ontologies lies in the definition of a language-independent lexical system capable of supplying a satisfactory backbone to all languages or at least to those in the system. A second option is to use a human language as the interlingua. This has been done in the DLT system (Schubert, 1992), which adopts Esperanto for its interlingua. A major drawback of this approach is that it forces an excessive dependency on the lexicon and conceptual structures of one of the languages involved. A third possibility

is to link the languages through a set of concepts, which form a superset of all concepts encountered in the different languages involved. The advantage of this solution is that there is no complex semantic structure that needs to incorporate the complexity of all languages involved. Such an unstructured index is used as a linking device in EuroWordNet (Vossen, 1998).

EuroWordNet is a multilingual database containing several monolingual wordnets which are structured along the same lines as Princeton WordNet (Miller et al., 1990; Beckwith et al., 1991; Fellbaum, 1998). WordNet is an on-line lexical reference system for English whose design is inspired by psycholinguistic theories of human lexical memory. It is an attempt to model the lexical knowledge of a native speaker of English. Information in WordNet is organised around logical groupings called synsets. Each synset consists of a list of synonymous word meanings between which basic semantic relations are expressed, for example, hyponymy (*car – vehicle*), meronymy (*bicycle – wheel*) and cause (*kill – die*).

In EuroWordNet, wordnets have been constructed for English, Dutch, German, Spanish, French, Italian, Czech and Estonian. In addition to the relations between synsets, the language-internal relations, each synset in EuroWordNet is also linked to an InterLingual Index (ILI) of concepts, thus forming a multilingual lexical database. The ILI is an unstructured list of WordNet synsets extended with any other concept that is needed to establish precise equivalence relations across synsets in the EuroWordNet database. Although the ILI itself is not structured in terms of semantic relations between concepts, two ontologies are linked to ILI records:

- a top-concept ontology, a hierarchy of a common core of concepts such as *Substance*, *Object*, *Experience*.
- an ontology of domain labels, e.g. 'sports', 'water sports', 'winter sports', 'military', 'hospital'.

These ontologies enable a user to customise the database with semantic features without having to access the language-internal relations of each wordnet. The ontology of domain labels can also be used to extract sublanguage vocabularies. The two ontologies are shared by all the languages in the EuroWordNet database. This means that there is some hierarchically structured semantic sharing. A schematic overview of the EuroWordNet architecture is given in Figure 2.5.

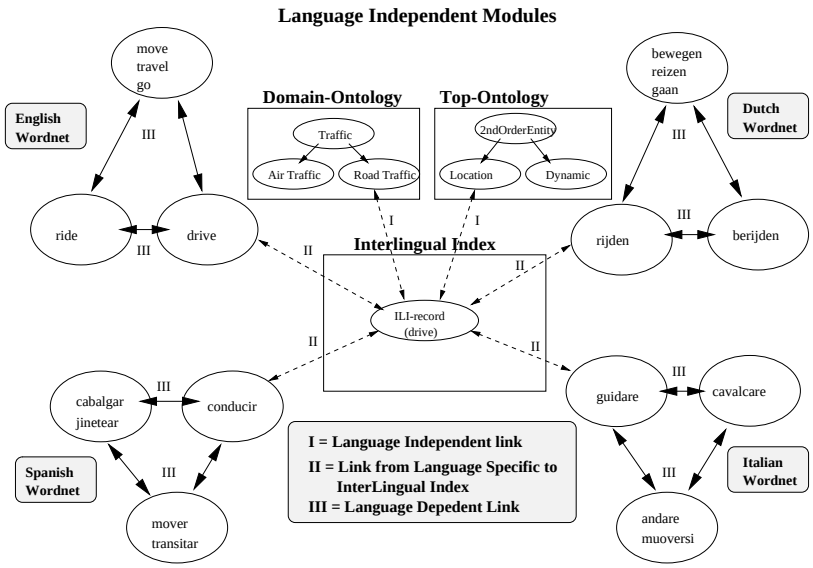


Figure 2.5: EuroWordNet architecture (Vossen 1998, p.80)

2.4 Inheritance-Based Approaches to Multilingual Lexicons

The multilingual lexicons that have been discussed so far, focussed on the sharing of semantic information. However, as the examples below show, languages also possess similarities at other levels of linguistic description – syntax, morphology, phonology, etc.

Phonology

Examples of phonological similarities are often found between words which come from a common origin. The table below illustrates this for three West Germanic languages – Dutch, English, and German[3].

English	*bed*	/bEd/	*rib*	/rIb/	*hand*	/h{nd/	*cat*	/k{t/
Dutch	*bed*	/bEt/	*rib*	/rIp/	*hand*	/hAnt/	*kat*	/kAt/
German	*Bett*	/bEt/	*Rippe*	/rIp@/	*Hand*	/hant/	*Katze*	/kats@/

[3]The transcriptions are taken from CELEX (Baayen, Piepenbrock, and van Rijn, 1995) and use the SAM-PA phonetic alphabet (Wells, 1987; Wells, 1989; Wells, 1995). Throughout this thesis, we use the SAM-PA phonetic alphabet for transcriptions.

Phonological similarities may also arise between words due to language contact. Languages borrow words from each other. Although borrowed words generally undergo some changes to integrate them in the phonological system of the target language, many phonological similarities remain. An example of this is the Russian word for 'trousers' *brjuki* /br'uk'i/ which was borrowed from Dutch *broek* /bru:k/.

Morphology

An example of shared morphology can be found in the subregular verbs in Dutch, English, and German. These three languages contain a set of subregular verbs which exhibit alternations which are very similar in all three languages. Compare, for example, the forms of the verb *sing*:

English	s**i**ng	s**a**ng	s**u**ng
Dutch	z**i**ng	z**o**ng	gez**o**ngen
German	s**i**ng	s**a**ng	ges**u**ngen

Syntax

An example of syntactic similarities between languages can be found at the level of subcatgorisation frames. Especially in related languages, the subcategorisation frames of verbs often exhibit identical argument slots and similar, if not identical, argument types as is the case for the verb *see* in English and Dutch (Heid and Krüger, 1996, p. 12).

`[PERCEIVER non-intentionally] see [actual entity PERCEIVED]`

English	*She saw tears in his eyes*
Dutch	*Zij zag tranen in zijn ogen*

Another example of a syntactic property which is shared by many languages is pro-drop, which means that a pronominal subject may be phonetically null in tensed sentences. Pro-drop occurs for example in Serbian/Croatian/Bosnian and Italian.

Serbian/Croatian/Bosnian	čitaš *You are reading*
Italian	canta *He/She is singing*

All those similarities could be captured in a multilingual inheritance network and, as noted by Cahill and Gazdar (1999b), this could contribute significantly to the robustness, maintainability, and extensibility of multilingual NLP systems.

First, a multilingual inheritance architecture offers a more economical encoding of lexical information just as inheritance lexicons in general. As information is stated only once, inheritance lexicons provide the benefit of reduced redundancy and therefore a more concise and transparent storage.

Second, there is the benefit of improved extendability both within languages and to include other, related, languages. It might be possible to add new languages to a lexicon by defining them by difference to related languages already available in the lexicon. For example, Afrikaans could be defined by reference to Dutch.

Third, a multilingual inheritance architecture offers improved robustness. It provides a more intelligent approach to lexical incompleteness. By exploiting default information from both the source and the target language, together with information about the default commonalities across those languages, it may be possible to deduce sufficient information about a missing lexical item via information which is available in the lexicon. For example, the German word for *forbid* could be deduced from the fact that the English verb *bid* translates as *bieten* and that verbs beginning with *for* in English generally begin with *ver* in German. This example is taken from Cahill and Gazdar (1995, p.175).

Finally, a multilingual inheritance lexicon may provide a formal account of how languages have diverged from their common origin. This may be of interest from a historical perspective, but it is not our concern here.

The projects below explore these ideas.

2.4.1 Kameyama's multilingual unification grammar for nominal expressions

Kameyama (1988) describes a prototype shared grammar for simple nominal expressions in Arabic, English, French, German and Japanese which is implemented in categorial unification grammar. Typologically non-related languages were chosen to demonstrate the feasibility of a shared syntactic rule base for dissimilar languages. In Kameyama's grammar a shared component is created by the process of **grammatical atomization**, which breaks grammatical assertions, expressed

as feature structures, up into smaller independent parts in order to reveal cross-linguistic invariant primitives. These shared primitives are prefixed with SG and can be due to universal, typological, genetic or areal bases. Language-specific templates are formed in this grammar by inheriting particular combinations of shared primitives. They are prefixed with AR(abic), EN(glish), FR(ench), GE(rman), and JA(panese).

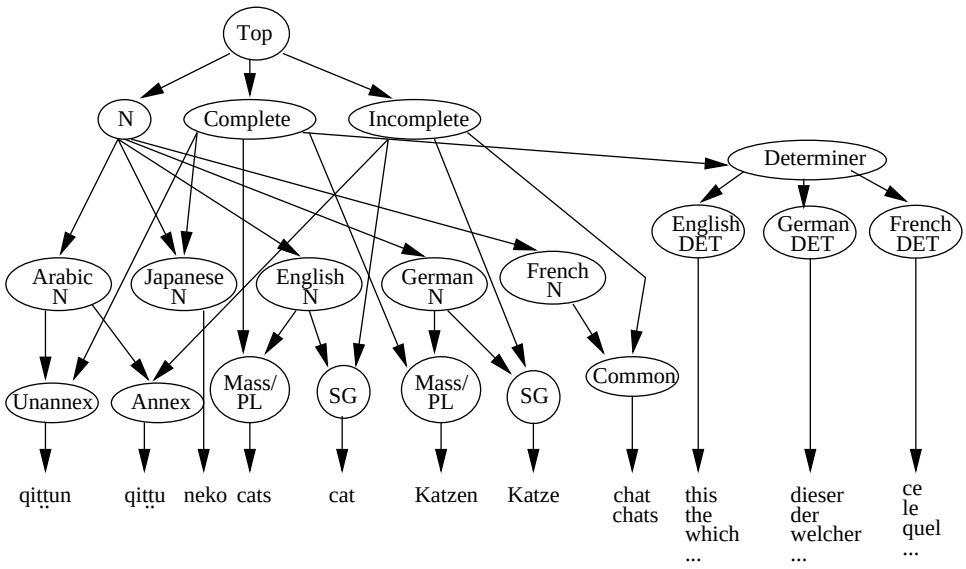


Figure 2.6: Shared Grammar from Kameyama (1988, p.195)

Kameyama's grammar is an example of a non-parameterised monotonic multiple inheritance network. Figure 2.6 shows a grossly simplified shared lattice of the kind that we might find in Kameyama's grammar. There is a universal notion `N(ominal)` which applies to all 5 languages under consideration. This common notion is part of the definition of `N` of each language by inheritance. In addition, there are two general categories `Complete` and `Incomplete` which apply to subsets of those. Complete means that the nominals can be used as subjects or objects as *cats* in *I saw cats*, while Incomplete means that they cannot be used as such (e.g. *cat* in **I saw cat*). Determiners in English, German, and French make incomplete nominals complete and therefore the Determiner definition in these languages inherits from Complete. Lexical items in the different languages are defined by multiple inheritance of the relevant assertions.

The reality is more complex than shown in this picture. To be able to share information between the languages of the test set, grammatical assertions had to be broken up into very small parts, increasing the complexity of the internal

structure of the grammar rules as well as the complexity of the validation and maintenance. Bateman (1997, p.43) assumes that this extreme 'atomization' is due to the choice of typologically non-related languages.

Kameyama's work was part of a long-term multilingual project launched at MCC (Slocum and Justus, 1985). Unfortunately, no more papers were published on the multilingual grammar.

2.4.2 PolyLex

Contrary to Kameyama (1988), the PolyLex project focuses on related languages. It defines a trilingual hierarchical lexicon for Dutch, English, and German sharing morphological, phonological and morphophonological information between them. The multilingual hierarchy is constructed in PolyLex by taking monolingual hierarchical lexicons for Dutch, English, and German and creating a parallel hierarchy which contains what the monolingual hierarchies have in common. An example of the resulting kind of multilingual inheritance network is given in Figure 2.7. This picture describes a lexicon fragment for Dutch, English, and German nouns. It consists of four interacting hierarchies, three language-specific hierarchies and one common hierarchy which contains the information that the language-specific hierarchies have in common. The network supports multiple inheritance which means that individual lexical entries can combine information unique to a particular language with information common to Dutch, English, and German. In PolyLex the language-specific hierarchies do not inherit information from other language-specific hierarchies. They can only inherit from the common hierarchy[4]. The solid lines in the figure, indicate inheritance of morphological information, and dashed lines indicate inheritance of phonological information. For illustrative purposes, only a small proportion of the inheritance links which actually exist both within and between the four hierarchies have been included. For example, the figure does not include links that exist between language-specific noun classes (e.g. `Noun_Dut`) and common noun classes (e.g. `M_Noun_1`). The PolyLex lexicons are implemented in DATR.

[4]Cahill and Gazdar did not want to make the language-specific hierarchies dependent on each other.

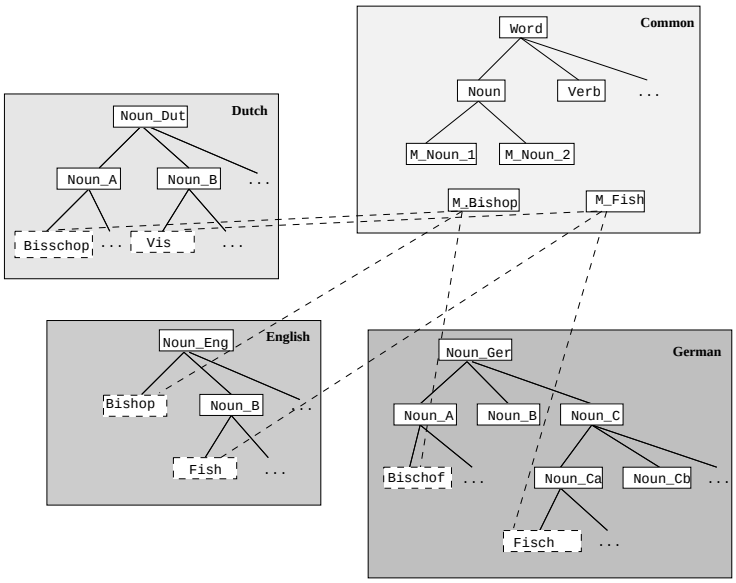


Figure 2.7: Multilingual Inheritance Hierarchy (Cahill and Gazdar, 1999b, p.14)

As mentioned, the PolyLex project focuses on the sharing of morphological, phonological, and morphophonological similarities between languages and there is no necessity for the words in question to share their semantics. In the PolyLex lexicon, words can inherit common phonology and morphology without sharing semantics. Compare English *keen*, Dutch *koen*, and German *kühn*. The German and Dutch words mean *brave* which is the ancestral meaning of *keen*. But this sense died out by the 17th century, and has been replaced by the meanings familiar today, such as 'eager' and 'sharp'. Thus, these three words come from the same root form historically, but the English *keen* has undergone semantic drift which has left it more distant from its cognates in meaning than in phonology, morphology. In PolyLex, the morphological and phonological similarities between *keen*, *koen*, and *kühn* will be shared.

The PolyLex architecture is what we call a non-parameterised approach to multilingual lexical representation. The multilingual lexicon consists of a set of monolingual inheritance lexicons plus a set of hierarchies capturing what the monolingual hierarchies have in common. Depending on the number of languages involved and their relatedness, several layers of subhierarchies can be introduced in the multilingual inheritance network. For example, if there are four languages, it may be possible to have three sets of shared hierarchies, one representing the information shared by two of the languages, another representing the information shared by

the other two languages, and one containing the information that is shared by all four languages in the lexicon. Subhierarchies were not used in the PolyLex project as the three languages were so closely related.

2.4.3 GREG

The most recent project in this area of research is GREG (Kilgarriff, Cahill, and Evans, 1999), which proceeded partly in parallel with this thesis. In GREG a multilingual valency lexicon was constructed for a set of 1000 Georgian verbs and their Russian, English, and German counterparts. To this end, GREG used a multilingual framework based on PolyLex, extending it in particular by allowing the lexicons to refer to each other. That is, in GREG the language-specific hierarchies can inherit directly from each other.

2.5 Other related work

This section discusses a few projects which are not about lexicons but could be considered complementary to the research described in this thesis in the sense that they try to capture linguistic similarities other than semantics between related (and unrelated) languages. We start with a brief discussion of a multilingual phonological grammar developed by Coleman et al. (1996). Then we describe KPML's multilingual development architecture and finally we say something about core grammars.

Coleman's parameterised multilingual phonological grammar

Coleman and his collaborators (Coleman et al., 1996) discuss a multilingual phonological grammar for the 'all-prosodic' IPOX synthesis system[5], which was developed on the basis of work done for Tashlhit Berber, Urdu, Dutch, and English. They define a declarative, constraint-based grammar which consists of a universal core containing the rules that apply to all languages, a set of parameterised rules defining dimensions along which languages may differ systematically, and language-specific data and rules. The focus of their work is on syllable-internal

[5]See `ftp://chico.phon.ox.ac.uk/pub/ipox`.

structure including phenomena such as syllable weight. In their model, a particular language is generated by setting a number of parameters and adding language-specific details. An example of parameter setting in their grammar can be found at the level of segments (Coleman et al., 1996, p.70). It is generally assumed that voiceless, unaspirated stops represent the unmarked case cross-linguistically and that distinctively voiced and aspirated stops are marked options. In their grammar this is expressed by two default parameter settings, `VoicedStops = No` and `AspiratedStops = No`. In order to process a language which includes voiced stops and/or aspirated stops, these defaults must be changed. For example for English, the value for `VoicedStops` must be changed to `yes`. Parameterisation is also used in our multilingual models. However, in our parameterised models, a language is not generated by setting a number of parameters, but information is made language-specific by using a language parameter.

KPML's multilingual development architecture

The KPML project (Bateman, 1997) is still ongoing. KPML (Komet Penman Multilingual) provides a development environment for multilingual lexical resources that are suitable for language generation. It is primarily concerned with capturing formal near-universals at the syntactic level (e.g. the declarative/interrogative distinction) and allows the large-scale development of resources which serve as a starting point for further application-specific customisation. In the KPML multilingual development environment, linguistic descriptions are made multilingual by conditionalising the linguistic objects for language, similar to the way that we use a language parameter in our parameterised models. This way, the resources are divided into partitions which hold for subsets of languages. The information that is valid only for one particular language is then the collection of specifications within all the partitions that mention that language. This model does not imply the existence of a universal core as in Coleman's multilingual phonological grammar.

KPML's position towards multilinguality is guided by two competing goals: **integrity** and **integration** of linguistic resources. Integration of different languages means that commonalities should be separated from particularity so that they can be reused. On the other hand, each individual language should preserve its integrity so that it can be used, maintained, and developed separately (Bateman,

1997, p.24,25). This way KPML supports a wide range of positions towards multilinguality, ranging from independant development for individual languages through to 'transfer comparison' where one of the languages is taken as the starting point for the description of another.

Core Grammars

Recently, there has been an increase in the development of so called core grammars. The term refers to the development of a base which is intended as the starting point for grammar development in different related languages. Pinkham (1996), for example, describes the development of a multilingual core grammar for the Romance languages starting from an English phrase structure grammar. Instead of adapting the English grammar for each language separately, Pinkham undertakes a conversion which will be less language-specific in the long run. To this end, a Core-Romance grammar is developed by transforming the English grammar into at least two Romance languages and defining the parts which are used by the two languages.

A potential drawback of developing a grammar from already available resources for closely related languages is that the resulting grammar might be biased towards the language it was originally developed for.

2.6 Summary

In this chapter, we have discussed various approaches to multilingual lexicons ranging from monolingual lexical resources linked at the level of semantics only (mainly MT lexicons) to lexicons sharing information at different levels of linguistic description. In this thesis, we are most interested in the second approach. We use an inheritance-based formalism and explore how such a formalism can be used to construct a multilingual lexicon in which information can be shared at different levels of linguistic description, building on the experience gained in the multilingual projects described in this chapter.

Chapter 3

Multilingual Architectures

3.1 Introduction

This chapter discusses the organisation of multilingual inheritance lexicons. First, a brief overview of inheritance techniques as used in monolingual settings is given in Section 3.2, focussing in particular on inheritance in DATR in Section 3.3. Section 3.4 then considers several ways in which the monolingual inheritance techniques can be extended to the multilingual case and the architectures for multilingual lexical representation that we explore in this thesis are defined.

3.2 Inheritance-based formalisms

Inspired by the idea of semantic networks in AI, inheritance networks provide a flexible means of representing lexical information. They structure the lexicon as a hierarchy in which information is pushed as far up the hierarchy as it can go capturing as many generalisations as possible. To this end, the lexicon includes objects corresponding to classes of words (e.g. nouns, verbs, intransitive verbs). These classes contain information associated with all words in that class and that information is propagated to others by inheritance.

Various kinds of inheritance networks have been proposed (Daelemans and (eds.), 1992). An inheritance network can be **monotonic** or **non-monotonic**. An inheritance network is **monotonic** if each node inherits all properties associated with

a parent node. An example of such a network is presented in Figure 3.1[1]. Here, the verb *hate* inherits all properties from transitive verb which inherits all properties from verb. Thus, its syntactic category is verb and its past participle ends in *-ed*. Monotonicity specifies that node-specific information only ever adds to the inherited information. It may not contradict or override it. If, on the other hand,

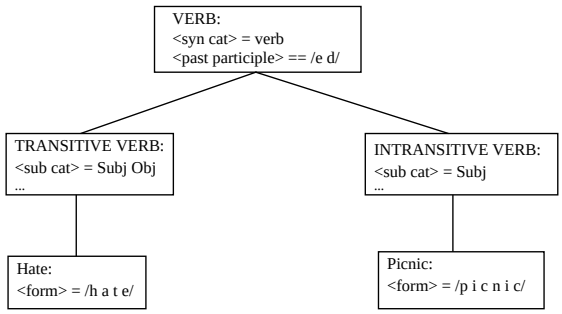


Figure 3.1: Monotonic Single Inheritance Network

information that is attached to a node takes precedence over information which is inherited from a parent, we speak of **non-monotonic** inheritance. An example of a non-monotonic inheritance network is given in Figure 3.2. In this network, the past participle information which is specified at the *beat* node overrides the past participle information inherited from the top of the hierarchy, resulting in a past participle *beaten* rather than *beated*. Non-monotonic inheritance is also known as **default inheritance**.

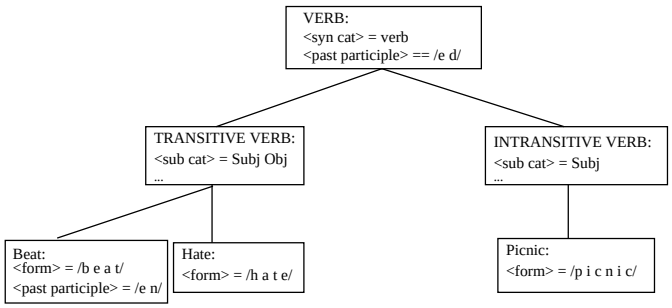


Figure 3.2: Non-Monotonic Single Inheritance Network

The inheritance networks we have seen so far were all **single** inheritance networks. This means that each class in the network inherits from at most one superclass. It

[1]The examples in this section are based on examples of Daelemans, De Smedt and Gazdar (1992).

is also possible that the inheritance network allows nodes to inherit from more than one parent. This is called **multiple** inheritance. Multiple inheritance networks can be monotonic or non-monotonic. An example of a monotonic multiple inheritance network is given in Figure 3.3. Here the verb *beat* is added to the monotonic

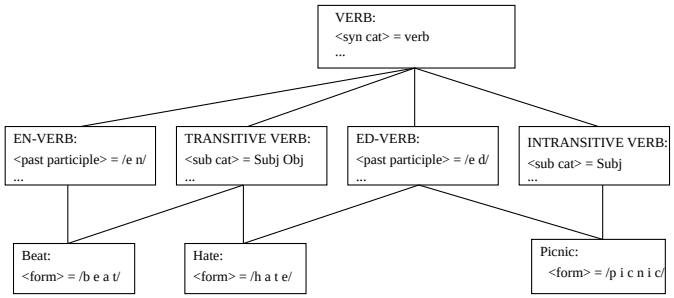


Figure 3.3: Monotonic Multiple Inheritance Network

inheritance network of Figure 3.1. The resulting hierarchy describes the same facts as the non-monotonic inheritance hierarchy of Figure 3.2. An example of non-monotonic multiple inheritance is given in Figure 3.4.

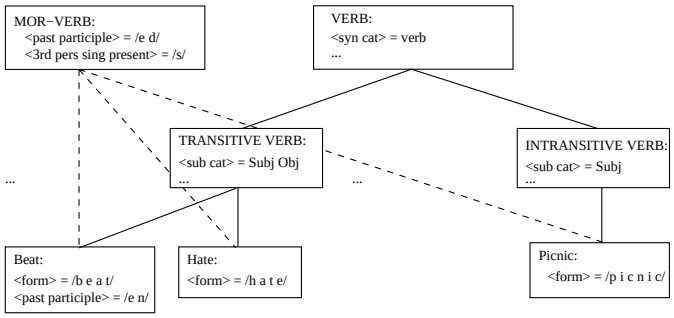


Figure 3.4: Non-Monotonic Multiple Inheritance Network

The problem with multiple inheritance is that it introduces a potential source of conflict: if there is more than one superclass to inherit from, the inherited information might be in conflict. Broadly speaking, two strategies have been adopted to avoid this issue, i.e. orthogonal inheritance and prioritized inheritance. **Orthogonality** means that a node cannot inherit the same information from more than one parent. For example, it is possible to inherit morphological properties from node `A` and phonological properties from node `B`, but it is not possible to inherit morphological properties from both node `A` and `B`. Another strategy is to order the parents: the first parent in the ordering that is able to supply the property

wins and contradiction is thus avoided. This is known as **prioritised inheritance**. In the Figures 3.3 and 3.4 conflict is avoided by splitting the inherited information such that verbs inherit their syntactic properties from the `TRANSITIVE` or `INTRANSITIVE` verb classes, and their morphological properties from the classes `EN_VERB` or `ED_VERB` or `MOR_VERB`.

Monotonic single inheritance networks are not really very well suited to capture the type of nested generalisations with exceptions that natural languages exhibit. Therefore, NLP systems generally use multiple inheritance, default inheritance or a combination of the two. Examples of projects using monotonic multiple inheritance are Kameyama's multilingual unification grammar (Kameyama, 1988) and the Core Language Engine (Alshawi, 1992). The advantage of using default inheritance is that it is good for capturing phenomena of blocking and for overriding regularities by subregularities (Calder, 1989; Briscoe, Copestake, and Lascarides, 1995). Examples of formalisms using both default and multiple inheritance include ELU (Russell et al., 1992) and DATR (Evans and Gazdar, 1996). ELU uses prioritised multiple inheritance. DATR, on the other hand, was designed to facilitate orthogonal multiple inheritance analyses, but it can also be used to define prioritised inheritance networks (Evans, Gazdar, and Moser, 1993).

DATR is a very general language for lexical description that can be used with a) any linguistic formalism/theory which can be encoded in terms of attributes and values, e.g. HPSG, LTAG; b) any domain of linguistic description – phonology, orthography, morphology, syntax, and semantics; c) any language – DATR fragments exist for a wide variety of languages, e.g. Arabic, Czech, Gikuya, Polish to list a few[2]. Different implementations of DATR exist. The best known and most widely available are Sussex/Brighton DATR[3], Gibbon's ZDATR[4] and Kilbury's QDATR[5]. In this thesis we will use Sussex/Brighton DATR to encode the multilingual architectures. In the next section, we discuss how inheritance works in DATR.

[2]An extensive archive of DATR fragments is available on the DATR webpages `http://www.datr.org`.

[3]See the DATR webpages.

[4]See the ILEX/DATR web site `http://coral.lili.uni-bielefeld.de/DATR/`.

[5]See the QDATR homepage `http://www.phil-fak.uni-duesseldorf.de/sfb282/B3/qdatr.html`.

3.3 Inheritance in DATR

DATR uses inheritance within feature structures to express inheritance relations. Paths in DATR can be conceptualised as specifications of increasing levels of detail because of DATR's **'definition by default'** mechanism. This mechanism allows a DATR statement to be applicable not only for the path specified in its left-hand-side, but also for any rightward extension of that path[6] for which no more specific statement exists. This can be illustrated with the present tense verb morphology in English. The form of present tense verbs in English is dependent on person and number, but for most verbs all forms are equal to the root except for the third person singular. This can be defined as follows in DATR:

```
VERB:
   <mor present> == root
   <mor present sing third> == root s.
```

Assuming that `number` has the values `sing` and `plur` and `person` has the values `first`, `second`, `third`, this definition implicitly defines:

```
VERB:
    <mor present sing first> == root
    <mor present sing second> == root
    <mor present sing third> == root s
    <mor present plur first> == root
    <mor present plur second> == root
    <mor present plur third> == root
```

Thus a short path states a broad generalisation (e.g. <`mor present`> states that all present tense forms have some value). A longer path increases the specificity of the generalisation (e.g. <`mor present sing third`> is just the third singular present tense forms). The most general path is the empty path, which is the leading subpath of every path, and as such acts as a 'catch all'. For example, in the following fragment `TRANSITIVE` inherits all information from `VERB` except for paths and their extensions defined under `TRANSITIVE` which is here the subcategorisation information.

[6] **Path extension**: a path `P2` extends a path `P1` if and only if all the attributes of `P1` occur in the same order at the left hand end of `P2` (so <`a1 a2 a3`> extends <>, <`a1`>, <`a1 a2`> and <`a1 a2 a3`>, but not <`a2`>, <`a1 a3`>) (Evans and Gazdar 1996, p.171).

```
VERB:
   <syn cat> == verb.
   <mor present> == root
   <mor present sing third> == root s.

TRANSITIVE:
   <> == VERB
   <sub cat> == subjobj.
```

In the remainder of this thesis, we will adopt the following terminology. We will use the term **inheritance hierarchy** to refer to the node hierarchy in an inheritance network (e.g. the `VERB-TRANSITIVE` hierarchy in our example), and we will use the term **feature tree** to refer to ordered feature structures (e.g. `tense`, `number`, and `person`).

3.4 Multilingual Inheritance

The previous section discussed the main characteristics of inheritance-based representation. This section explores how inheritance can be used in a multilingual context. A multilingual inheritance network aims to capture generalisations within and across languages. In order to capture cross-linguistic generalisations, the inheritance network needs to have some kind of means to indicate that parts of the network are valid for more than one language. Before we discuss how this can be done, we first need to consider the kind of relationships between languages that can be modelled in the lexicon.

The multilingual lexicon may capture only generalisations across all the languages represented in the lexicon (i.e. it uses a flat language typology) or the lexicon may divide languages into classes (such as Germanic or Romance) based, for example, on the genetic relationships between languages and capture generalisations between language classes and individual languages, or the lexicon can model dialects of languages or diachronic variants. The result is a more or less complex typology of languages and a corresponding structure in the lexicon with different parts of the lexical representation associated with different points in the typology.

The relationships between languages are very similar to the relationships that were modelled using inheritance in a monolingual inheritance lexicon. For example, a

lexicon for Dutch, English, Spanish, and French might use the language typology represented in Figure 3.5. Information which is shared by all those languages

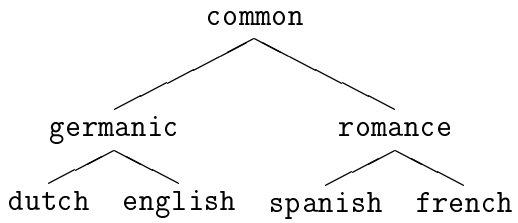


Figure 3.5: A language typology

will be associated with `common`. Information which is specific to the Romance languages will be associated with `romance` and information which is specific to French or Spanish will be associated with respectively `french` or `spanish`. In this multilingual lexicon, it might be reasonable to suppose that `french` and `spanish` inherit from `romance`, `dutch` and `english` inherit from `germanic`, and both `germanic` and `romance` inherit from `common`. The inheritance relations could be monotonic or non-monotonic.

This sketches a broad picture of how the inheritance relations between languages might be organised in a multilingual inheritance lexicon. We now turn to how parts of the lexicon can be linked to this language typology. Evans (1996) distinguishes two approaches which he calls parameterised and non-parameterised.

In a **non-parameterised** model, the multilingual lexicon is constructed by taking a set of monolingual hierarchical lexicons and creating a parallel hierarchy containing what the monolingual lexicons have in common. The resulting structure for a multilingual lexicon with a flat language typology is illustrated in Figure 3.6. In this figure, we see that all three monolingual hierarchies have a `Word`, `Noun`, `Adjective`, and a `Verb` class. This shared information is captured in a shared hierarchy at the top of the inheritance network. The structure of a multilingual lexicon where languages are grouped into classes is given in Figure 3.7. In this figure, two of the three languages share the `N_a` subclass. This generalisation is captured by grouping these two languages together into a subhierarchy. Such a subhierarchy could correspond to a (sub)family of languages. Evans calls these networks non-parameterised because language is not explicitly used as a parameter. There is in principle nothing which ties a particular hierarchy to a particular language in the multilingual inheritance structure. Each hierarchy belongs to an

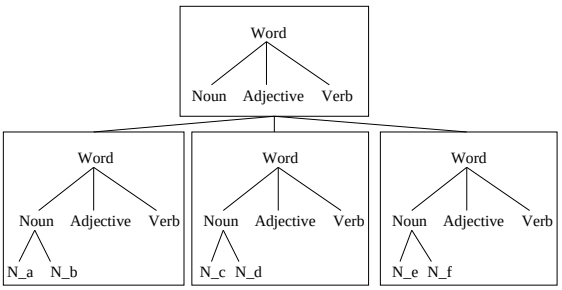


Figure 3.6: Non-parameterised multilingual inheritance hierarchy with a flat language typology

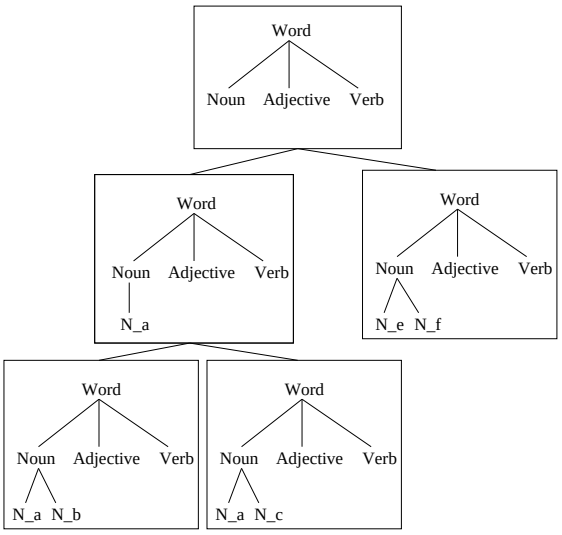


Figure 3.7: Non-parameterised multilingual inheritance hierarchy with subhierarchies

individual language or represents information shared by a set of languages, but nothing in the hierarchy tells you explicitly which language or languages are concerned – the knowledge of the different languages involved is in the user's head rather than in the theory.

Alternatively, it is possible to integrate all the languages represented in the lexicon into a single hierarchy and to use language as a parameter to indicate which parts of the lexicon are valid for which languages. Evans call this approach **parameterised**. A schematic illustration of a parameterised model is shown in Figure 3.8. The different boxes indicate which part of the hierarchy is valid for which lan-

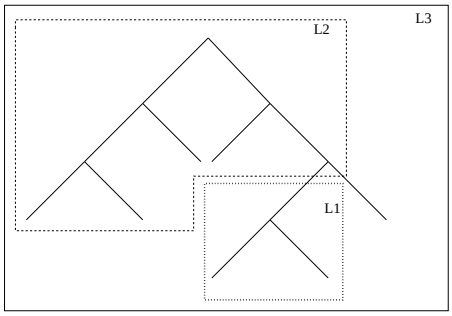


Figure 3.8: Parameterised multilingual inheritance hierarchy

guage. The whole hierarchy is valid for language `L3`, the dashed line indicates the part which is valid for language `L2`, and the dotted line indicates the part that is valid for language `L1`. Evans uses DATR for his multilingual inheritance architectures and parameterises linguistic descriptions by introducing language parameters in DATR's main feature theory. He suggests three possible models: the Micro-Features model, the Meta-Features model, and the Infinitesimal model.

Evans' proposals assume that a multilingual lexicon consists of the representations of more than one (monolingual) lexicon at the time. It is also possible to imagine a parameterised model in which the objects of description are not lexicons as such, but rather default inheritance descriptions of lexicons. A lexicon for a particular language can be compiled out from this multilingual metalexicon by querying the metalexicon for that particular language. We call this the MetaTheory model.

Below, a detailed description is given of these different varieties of parameterised and non-parameterised models.

3.4.1 Non-parameterised multilingual inheritance

We saw above that in a non-parameterised model, a multilingual lexicon is constructed by taking, conceptually at least, monolingual hierarchical lexicons for each of the languages and creating a parallel hierarchy containing what the monolingual hierarchies have in common. Thus, if there are n monolingual hierarchies, there will be at least *n+1* hierarchies in the multilingual architecture. An example of such a multilingual architecture was given in Figure 3.6 and 3.7. There is in principle nothing in this model which ties a particular hierarchy to a particular language. This means that generalisations about language inheritance cannot be captured. The basic inheritance pattern within a language is repeated at every level in the parallel hierarchies, resulting in trees with roughly the same structure, but generalisations about language inheritance cannot actually be captured. Another consequence of not having a language parameter is that identifying the lexical entries of a particular language or the lexical entries for a single word in all the languages is essentially an arbitrary matter. Evans also calls the non-parameterised model, the Structure-Sharing model.

The Structure-Sharing model is essentially the model that has been used in the PolyLex (Cahill and Gazdar, 1999b) and GREG (Kilgarriff, Cahill, and Evans, 1999) projects. Figure 2.7 showed a fragment of a multilingual inheritance hierarchy as has been used in the PolyLex project. Here we give a specific example of the sharing of phonological information between lexeme nodes in Dutch, English, and German in PolyLex.

PolyLex assumes a contemporary phonological framework (Cahill and Gazdar, 1997) in which all lexical entries are defined as having a phonological structure consisting of a sequence of structured syllables, a syllable consisting of an onset (the initial consonant cluster, which might be split up into onset 1, onset 2, etc.) and a rhyme. The rhyme consists of a peak (the vowel) and a coda (the final consonant cluster, which might be split up into coda 1, coda 2, etc.). This structure is defined at the top of the hierarchy, and applies by default to all words. Only the relevant values for onset, peak, and coda have to be defined for each syllable at the individual lexical entries. The values `syl1` and `syl2` denote syllable position[7].

[7] Cahill and Gazdar (1999a, p.6) number syllable sequences from the right, to reflect the fact that Germanic morphology primarily involves suffixation. Reference to final syllables is thus more frequent than reference to the initial syllable and it is technically convenient to have a constant identifier for final syllables.

Table 3.1 contains the definition of the lexeme nodes for the words *Bishop* and *Fish* in all three languages. The nodes `M_Bishop` and `M_Fish` are not really lexical

	bishop	*fish*
Common	M_Bishop: <> == NOUN <phn syl1 onset> == S <phn syl2 onset> == b <phn syl1 peak> == O <phn syl2 peak> == I <phn syl1 coda> == p <phn root> == Polysyllable2 <phn root focus> == syl2 <sem type> == human.	M_Fish: <> == NOUN <phn syl1 onset> == f <phn syl1 peak> == I <phn syl1 coda> == S <sem type> == animal.
English	Bishop: <> == M_Bishop <mor> == Noun <phn syl1 peak> == @.	Fish: <> == M_Fish <mor> == Noun_0.
German	Bischof: <> == M_Bishop <mor> == Noun_umlautE <phn syl1 coda> == f <phn syl2 coda> == Null.	Fisch: <> == M_Fish <mor> == Noun_E.
Dutch	Bisschop: <> == M_Bishop <mor> == Noun_EN <phn syl1 onset> == s x.	Vis: <> == M_Fish <mor> == Noun_EN <phn syl1 onset> == v <phn syl1 coda> == s.

Table 3.1: Lexeme entries for *Bishop* and *Fish* (Cahill and Gazdar, 1999b, p.18).

entries. They are abstract nodes containing information shared by the majority of the forms in the different languages. Looking at the entry for *Fish* we see, that the English and German entries inherit all the information specified under `M_Fish`. Thus, the onset is an /`f`/, the peak is an /`I`/ and the coda is an /`S`/ resulting in /fIS/ in English and German. For Dutch, the values for the onset and coda need to be overridden to get the Dutch form /`vIs`/. The nodes `NOUN`, `Noun_EN`, etc. refer to different noun classes that are distinguished in the three languages.

Both PolyLex and GREG are non-parameterised. During the PolyLex project, Cahill and Gazdar experimented with the inclusion of a language feature. This work is discussed in Cahill and Gazdar (1996), which describes a multilingual lexical analysis of numerals in Dutch, English, and German. In this work a language feature is prefixed to the feature-value path. However, the value of this feature has to be set before compilation and cannot be changed once the lexicon has been compiled. Thus, setting the language feature to German, compiles the file for German and will output German numerals, setting it to Dutch, will do the same for Dutch, etc. This means that to get a different language, one has to change the value of the node which specifies the language feature and consequently one has

to change the theory. In the parameterised models discussed below, the language parameter does not have the be set before compilation. It is given as input to query the lexicon and by changing the value of this parameter output in a different language is generated.

3.4.2 Parameterised multilingual inheritance

Following Evans' proposals we focus on parameterised models in which language parameters are inserted in the feature tree. For the purposes of the present discussion, we assume that language parameters are organised in a tree structure based on the genetic relationships between languages such as the classification for the Germanic languages given in Figure 3.9. See Chapter 4 for more discussion of language classifications. Inheritance relations exist in this language tree as they do in the feature tree. Thus in the language tree in Figure 3.9, Dutch will inherit from Germanic West Continental West and Continental will inherit from Germanic West etc.

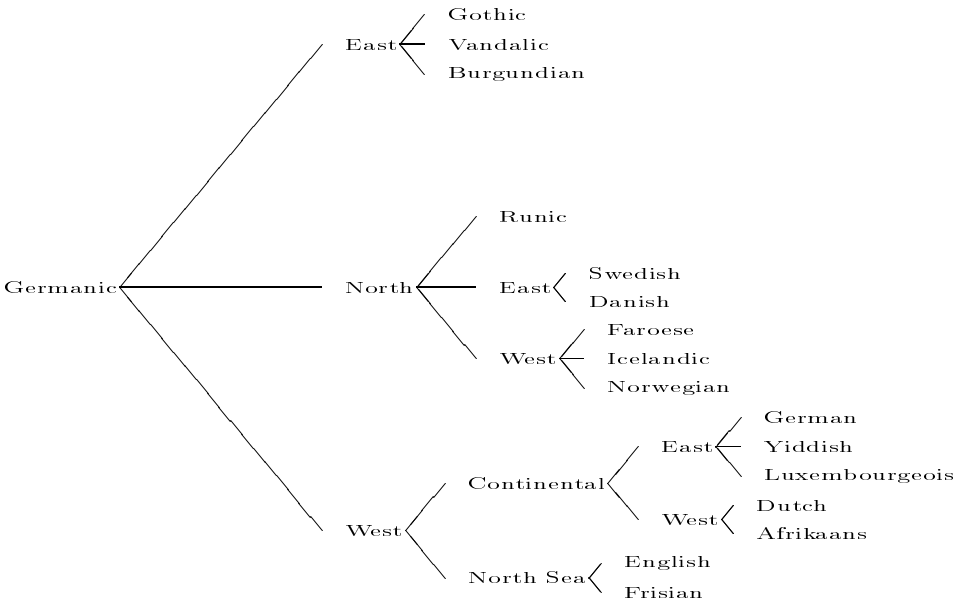


Figure 3.9: Classification of the Germanic languages

Thus a parameterised model consists of a feature tree for a particular structure (e.g. noun features) and a language tree representing the language typology. The

question that arises is how can these two trees be combined? In other words, where in the feature tree can language parameters be inserted and once language parameters have been inserted, how can inheritance be made to work in both the language tree and the feature tree. In DATR both the inheritance relations in the language tree and in the feature tree can be modelled using DATR's 'definition by default' mechanism. However, definition by default cannot be used in the feature and language trees at the same time. Evans suggests three models, which we will discuss below. First, the language tree is inserted at the bottom of the feature tree, the Micro-Features model. Second, the language tree is inserted at the top of the tree, the Meta-Features model. Third, the language tree can be inserted at any point in the feature tree, the Infinitesimal model.

The Micro-Features Model

In the Micro-Features model, language parameters occur at the bottom of the tree as is illustrated in Figure 3.10. Generally in inheritance networks, the lower the

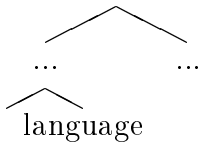


Figure 3.10: Illustration of the Micro-Features model

position in the hierarchy at which a property appears, the more exceptional it may be considered. Thus, the Micro-Features model is based on the assumption that variation between languages is exceptional rather than rule. It assumes one shared tree structure for all languages with only local, low-level variation occurring at the bottom of the tree.

An example of such variation is given in Figure 3.11 for the present tense verb morphology in Dutch, English, and German where all features and values in the tree are shared except for the actual realisation of the suffixes marking a particular form. The third person present tense suffix is a *-t* in Dutch and German and an *-s* in English.

Another example of minor variation between languages occurs when languages share the same features but have different feature-values. For example, nouns in two languages may both have the feature `number`, but in language 1, `number` has

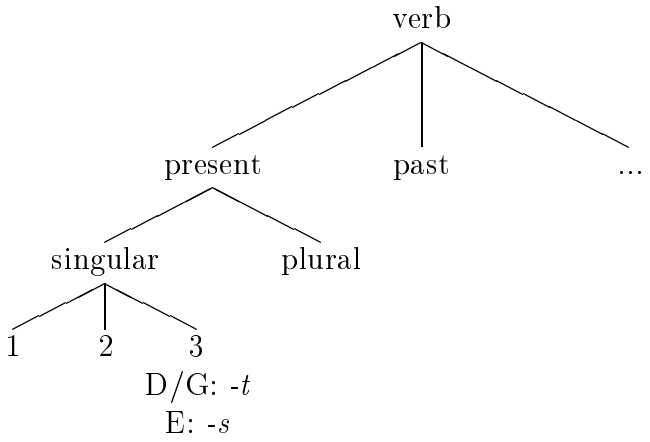


Figure 3.11: Feature tree for verb morphology in Dutch, English, and German

the values `singular` and `plural`, whereas in language 2 it has the values `singular`, `dual`, and `plural`.

Both these situations can be modelled in DATR using DATR's definition by default mechanism in the language tree.

However, generally, different languages have different feature trees and there are higher level language-dependent generalisations, even between closely related languages, that an adequate multilingual lexicon should be able to capture. This is illustrated below with the feature trees for the noun features in Dutch, English, and German. Nouns only inflect for `number` in Dutch and English, whereas they inflect for `number` and `case` in German.

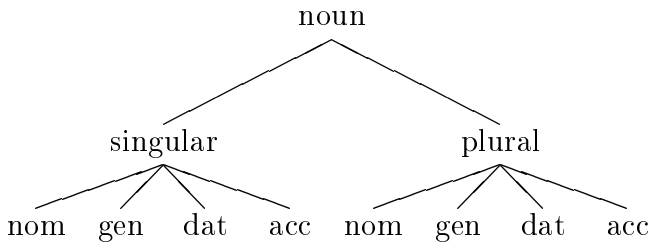


Figure 3.12: Feature tree for nouns in German

The Micro-Features model cannot deal with this situation. For the Micro-Features model, the feature tree has to be the same (i.e. have the same features, not necessarily the same feature-values) up to the point where language is inserted,

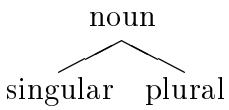


Figure 3.13: Feature tree for nouns in English and Dutch

which is completely at the bottom in the Micro-Features model. Thus the Micro-Features model cannot capture higher level generalisations such as that singular nouns in Dutch and German are subject to final devoicing whereas they are not in English. The applicability of the Micro-Features model is therefore limited and it will not be further considered as a viable option for constructing multilingual inheritance-based lexicons.

The Meta-Features Model

The Meta-Features model does the opposite of the Micro-Features model and language parameters occur at the top of the tree as is shown in Figure 3.14. The

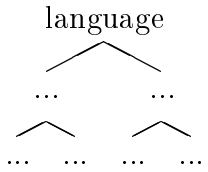


Figure 3.14: Illustration of the Meta-Features model

higher the position in the hierarchy at which a property appears, the more general it may be considered. Thus, the Meta-Features model is good at capturing higher level language-dependent generalisations such as nouns in one particular language have a property `x`, whereas nouns in general have a property `y`.

The Meta-Features model is not as restricted as the Micro-Features model which requires languages to have the same feature tree except for minor variations occurring at the deepest level. In the Meta-Features model, language variation can occur at the top of the tree and thus the tree structure does not have to be the same for all languages in the lexicon.

For example, the Micro-Features model could not capture that Dutch, English, and German have different feature trees for nouns. In the Meta-Features model,

this difference can be captured by inserting a language parameter at the top of the tree, resulting in two separate feature trees.

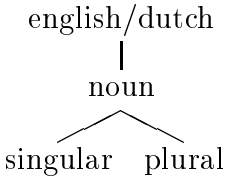


Figure 3.15: Feature tree for nouns in English and Dutch

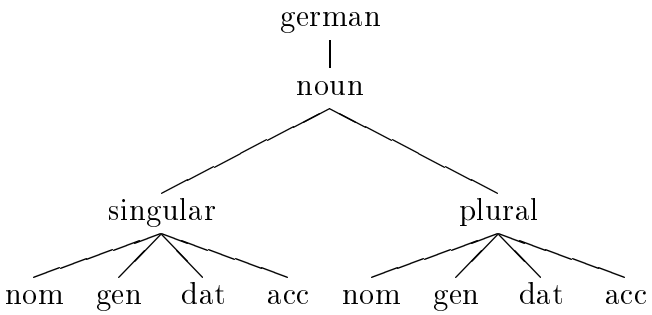
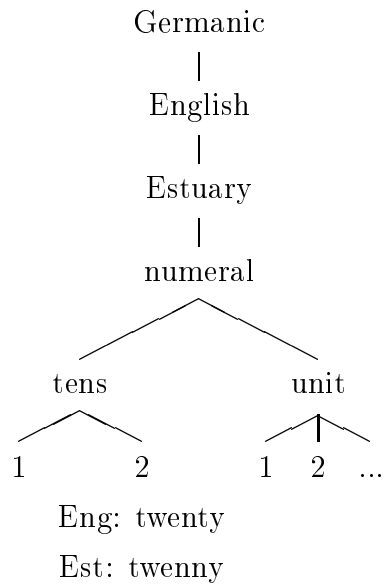


Figure 3.16: Feature tree for nouns in German

However, the Meta-Features model still does not allow us to capture the fact that Dutch and German singular nouns are subject to final devoicing whereas English nouns are not. In the Meta-Features model, the feature tree can either be completely the same or completely separate. Generalisations at intermediate levels cannot be captured.

As inheritance relations exist in both the language tree and the feature tree, the Meta-Features model is also good for expressing minor variations between languages. For example, adding a new dialect to the lexicon which is related to one of the languages already encoded in the lexicon, requires a change in the language typology, but it does not necessarily affect the feature tree. This is illustrated below with an extract of a feature tree for numerals in English and Estuary, a dialect of English (Evans, 1996).



Here, English and Estuary have the same feature tree with different values for <`numeral tens 2`>. Estuary will inherit all information that is specified for Germanic English, except for the value of <`numeral tens 2`> which is *twenny*.

The Infinitesimal Model

The Infinitesimal model combines the features of the Micro-Features model and the Meta-Features model. Language parameters can occur at the top or at the bottom of the tree or anywhere in between.

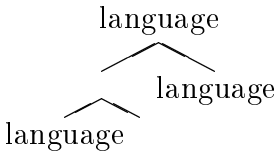


Figure 3.17: Illustration of the Infinitesimal model

Thus, language-specific characteristics can be captured at any level in the tree, completely at the top (to capture that nouns in language 1 behave differently from nouns in language 2), or completely at the bottom (to capture for example that singular nominative nouns in language 1 do something different from singular nominative nouns in language 2), or anywhere in the middle where one language behaves differently from the other(s). An example of the Infinitesimal model is shown below with a tree structure for the noun features for German and Danish. In Danish, nouns inflect for `number` and `definiteness`, whereas they inflect for

`number` and `case` in German. The different feature trees are integrated into one

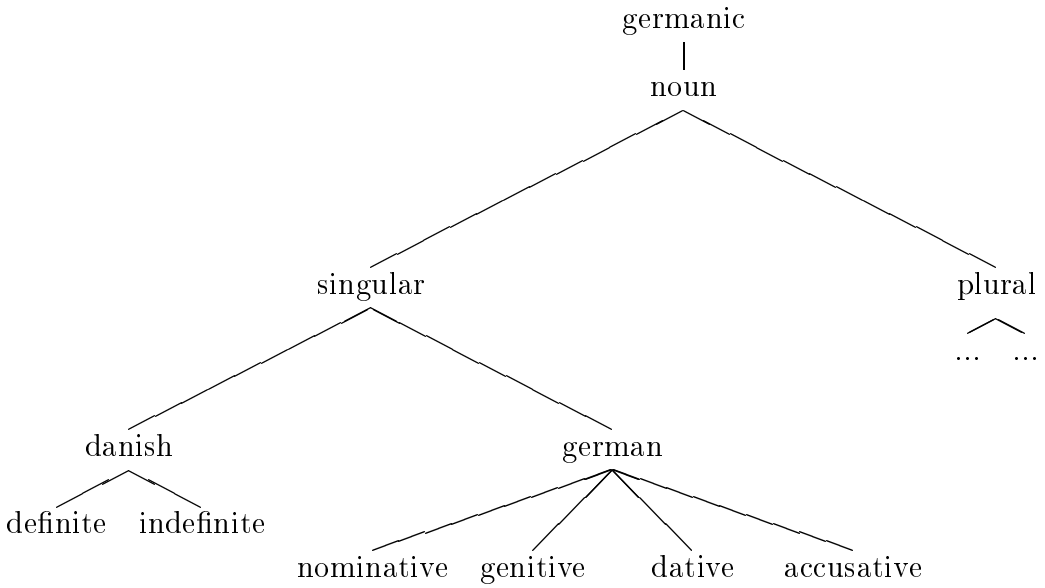


Figure 3.18: Parameterised tree structure for a subset of the Germanic languages

shared tree with a part that is specific to Danish and a part that is specific to German.

Potentially, the Infinitesimal model is the most powerful model, as it allows one to capture language variation anywhere in the hierarchy. The question that arises is do we want language variation anywhere in the tree or do we want language parameters to occur at particular places in the tree. We will return to this issue in the discussion of the implementation of the Infinitesimal model in Chapter 6.

The MetaTheory model

The last example of a parameterised model that we consider is the MetaTheory model. The MetaTheory model is conceptually different from the models we have discussed so far. The objects of description are not lexicons as such, but rather default inheritance descriptions of lexicons. The multilingual lexicon is a metalexicon which consists of a universal metatheory of the formalism that is used (e.g. DATR), which is the same for all metalexicons using this formalism and a particular metatheory whose theorems comprise object level theories of the different languages covered. The similarities between the languages in the lexicon are

captured in the particular metatheory. The object level theory of a particular language can be compiled out by asking the multilingual lexicon for the object theory of that language. The MetaTheory is in a way quite like the Structure-Sharing model. Language does not occur in the object theories of the particular languages, but in this case it is inserted in the metatheory.

The MetaTheory model is illustrated in Figure 3.19 for Dutch, English, and German. In this figure, the universal metatheory describes the DATR syntax. The

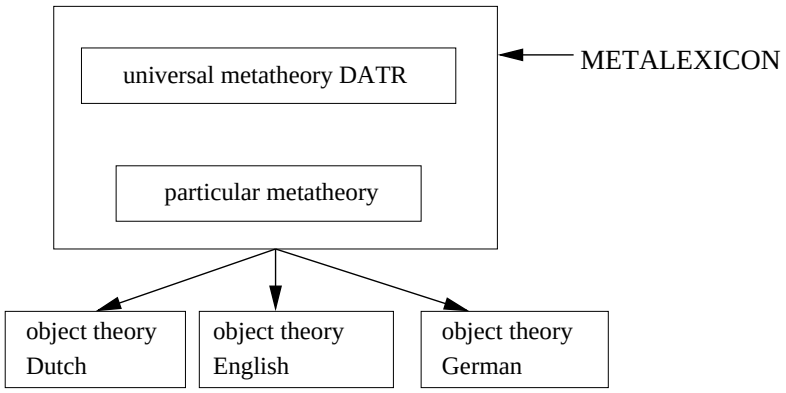


Figure 3.19: Example of the MetaTheory model

particular metatheory describes the Dutch, English, and German lexicons using metadefinitions which capture the similarities and differences between the monolingual inheritance descriptions of these languages. For example, let us take the lexeme *Fish* in Dutch (/`vIs`/), English (/`fIS`/), and German (/`fIS`/) as defined in Table 3.1. The metanode defining this lexeme will capture the fact that the value of the peak is shared by all three languages and that the values of the onset and the coda are shared by English and German, but it will also specify different values for the onset and coda in Dutch. The metalexicon can be queried for Dutch, English, and German, resulting in respectively a Dutch, English, and German object theory.

3.5 Conclusion

In this chapter, we defined the multilingual inheritance architectures that we explore in this thesis. In order to compare these different architectures, sample lexicons have been implemented in DATR. These fragments cover a small set of

nouns and adjectives in Dutch, English, Danish, and Icelandic. The results of the implementation are described in Chapter 6. Before turning to the results, we first discuss methodological issues such as data selection and information sharing in Chapter 4. Chapter 5 discusses the lexical description framework that we use.

Chapter 4

Methodology

4.1 Introduction

In order to compare the different architectures described in the previous chapter, sample lexicons have been implemented. This chapter discusses the methodological issues involved in the implementation of the sample lexicons. First, the selection of the **test data** is described. To make sure that the results are generally valid – i.e. the multilingual architectures are applicable to the vocabularies of whatever subset of languages is chosen – it is necessary to use a language sample that explores as much as possible the full range of forms and constructions that can occur in natural language. Section 4.2 describes the method that we used to select our language sample and discusses the selection of a subset of the vocabularies of the languages of the sample. A second issue concerns **information sharing**. That is, how do you decide which information can and should be shared between the different languages once the data has been selected and how can the information be shared. Section 4.3 describes the approach to multilingual information sharing adopted in this thesis. Section 4.4 looks into different **development strategies** that can be used to build a multilingual inheritance lexicon. The chapter concludes with an overview of **evaluation measures** for multilingual inheritance lexicons.

4.2 Data Selection

The selection of the test data involves two levels of sampling because of the multilingual nature of our research. First, a language sample has to be selected (i.e.

language sampling) and second, a subset of the vocabulary of the languages in this sample has to be chosen (i.e. **lexical sampling**). Section 4.2.1 discusses the language sampling and Section 4.2.2 the lexical sampling.

4.2.1 Language Sampling

In recent years, interest in cross-linguistic research has grown, and more attention has been paid to the quality of language samples, especially in language typology. New methods have been developed (Bell, 1978; Dryer, 1989; Perkins, 1989; Rijkhoff et al., 1993). We will follow Rijkhoff et al. to establish our language sample. The advantage of Rijkhoff's method is that it is the only one that can be used regardless of the classification that is chosen and the subject of research. Hence, it can also be used to sample one single language family, as will be done here.

Rijkhoff's method

The aim of Rijkhoff's method is to produce language samples in which the **differences** between individual languages are **maximal**. There are two ways to make sure that a language sample is genetically diverse. One is to take the variation across language families into account; the other is to consider the variation within individual language families (some are more diverse than others). Both factors are captured by Rijkhoff's sampling method. It is based on the following assumptions:

(i) the universe from which the sample is taken contains all known extant and extinct languages;
(ii) every language family should be represented by at least one member;
(iii) the number of languages that belong to the same language family should be proportional to the linguistic diversity in that particular language family;
(iv) the linguistic diversity (Diversity Value or DV) of a language family is determined on the basis of an objective measure.

The first component of the sampling procedure is relatively simple: i.e. every language family should be represented by at least one member – **minimal representation**. However, if this were the only rule, samples could never contain more than around 27 languages, the number of language families that exist. Thus, if we want a larger sample, languages belonging to the same language family will have to be chosen. To determine how to choose genetically related languages, Rijkhoff adopts a **proportional representation**, i.e. the number of languages by which

each language family is represented in a sample is proportional to the linguistic diversity within that language family.

Rijkhoff assumes that the linguistic diversity of a language family is reflected by the graph theoretic structure of its genetic tree. It is determined by the depth and the width of the genetic tree, i.e. the hierarchical structure of the different levels (generation), the number of nodes at each of these levels (parents), and finally the number of branches under each node (children). This measure is then used to determine how many languages of each language family should be selected, given any required sample size.

Rijkhoff provides the following method for computing the linguistic diversity of a language family. First, the genetic language tree has to undergo some preparatory transformations. The top and the bottom level of the tree are ignored (Figure 4.1). At the top level, the name of the language family is defined, which does not add

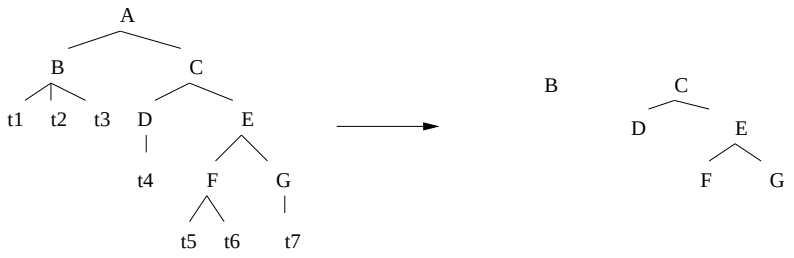


Figure 4.1: Tree transformation

extra information. At the bottom level, the names of the individual languages are specified. By disregarding the terminal nodes the influence of the actual number of languages which make up a language family is restricted and the weight is shifted entirely to the internal structure. This is important because numerically larger groupings are not necessarily more diverse than certain small families. Therefore, one should not rely too heavily on the absolute number of genetically related languages in an attempt to create maximum linguistic diversity in a sample.

After these transformations, the width of a language family can be determined at each intermediate level. To account for the fact that every separate branch adds to the width of the language family as a whole, all higher level preterminal nodes are extended to the deepest level of the representation of the tree. This is illustrated in Figure 4.2 where hyphens indicate extensions of higher level preterminal nodes. The width at some level is equal to the number of nodes at that level plus the

number of preterminal nodes above that level.

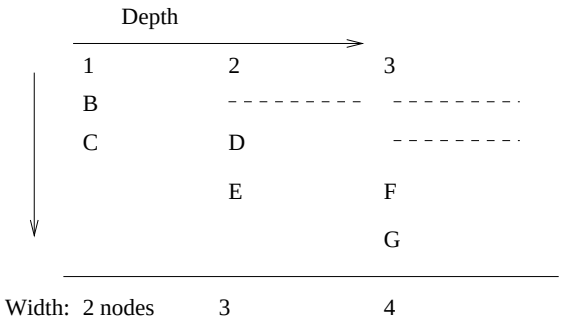


Figure 4.2: Width of the three intermediate levels of the BC subfamily

With this information, the contribution (C_y) of each intermediate level (`y`) to the linguistic diversity of a genetic tree can be computed by means of the following formula:

$$C_y \texttt{ = } C_x \texttt{ + ((n-x)/n} \times \texttt{ (}N_y \texttt{ - } N_x\texttt{))}$$

where `x` = `y` -1

C_y = contribution of intermediate level `y`

`y` = intermediate level

N_y = number of nodes on level `y`

`n` = the highest number of intermediate levels found in any language family

In other words, the contribution (C_y) is obtained by adding the contribution of the level immediately above level `y` (C_x) to the extra nodes of level `y` (as compared to the number of nodes at level `x`, i.e. N_y-N_x), the latter being multiplied by a fraction decreasing over the levels according to the row `n/n`, `(n-1)/n`, ..., `1/n`. By definition, the contribution of the first intermediate level (immediately below the top node) is equal to the number of nodes at that level (C_1 = N_1). The DV of a particular language family can now be calculated as the mean value of the contributions of all intermediate levels of this language family. By using the DVs as proportions, the number of languages that should be taken from each language family can be determined. Note that this formula assumes that high level splits in the tree are more significant in terms of variation than low-level splits – the contribution of the extra nodes at each deeper intermediate level is decreased by steps of `1/n`. The reason for this is that high-level splits have occurred earlier in

time than low-level splits, which means that two languages separated by a high-level split have had more time to develop into distinct languages than languages separated by a low-level split.

Sampling the Germanic Language Family

We will now use Rijkhoff's method to sample the Germanic language family. Due to limitations of time and feasibility, it was decided to limit the language universe to the Germanic language family, and a representative subset of Germanic languages has been chosen to test the different multilingual architectures. Sampling of the Germanic languages involved:

A. Choice of genetic classification

Different genetic classifications have been proposed for the Germanic language family: Voegelin (1966), Ruhlen (1987), Ethnologue (Grimes, 1996). The most recent classification, viz. the Ethnologue[1] has been used (Figure 4.3).

B. Determination of sample size

Rijkhoff's method states that each language family has to be represented by at least one member. The genetic tree of the Germanic languages shows that there are three subfamilies at the highest level. The minimal sample size is thus three. As the East Germanic family only has one member (and hence a linguistic diversity of 0), increasing the sample size will not affect the number of languages that is taken out of the East Germanic family. This will remain one. If we increase the sample size, then the extra languages will be taken out of the North and West Germanic families depending on their linguistic diversities. Consequently, we decided to increase the minimal sample size by two, resulting in a sample size of five.

C. Computation of the linguistic diversity

First the genetic tree of Figure 4.3 has to undergo some preparatory transformations resulting in the diagram represented in Figure 4.4. On the basis of this diagram[2], the linguistic diversity values can be computed.

Table 4.1 and 4.2 show the computation of the linguistic diversity for the West

[1] The same calculations were done with Ruhlen's classification. The results were the same.

[2] Note that in case the set of daughters of some node `N` consists of both non-terminal and terminal nodes (i.e. groups and individual languages), an extra preterminal node is inserted between node `N` and terminal node `t`. In Figure 4.4, the nodes `A`, `B`, `C` have been inserted under Low Continental West Germanic, which consists of the non-terminal `Dutch` and the terminals `GERMAN (LOW)`, `PLAUTDIETSCH`, and `SAXON (LOW)`.

Level	Nodes				Contribution
		C_x	+ ((n-x)/n	× (N_y-N_x))	
1	1				1
2	2	1	+ ((4-1)/4	× (2-1))=	1.75
3	4	1.75	+ ((4-2)/4	× (4-2))=	2
4	7	2	+ ((4-3)/4	× (7-4))=	2.08

DV = 5.83/4 = 1.7083

Table 4.1: Computation of DV of the West Germanic language family

Level	Nodes				Contribution
		C_x	+ ((n-x)/n	× (N_y-N_x))	
1	1				1
2	3	1	+ ((4-1)/4	× (3-1)) =	1.375

DV = 2.375/2 = 1.1875

Table 4.2: Computation of DV of the North Germanic language family

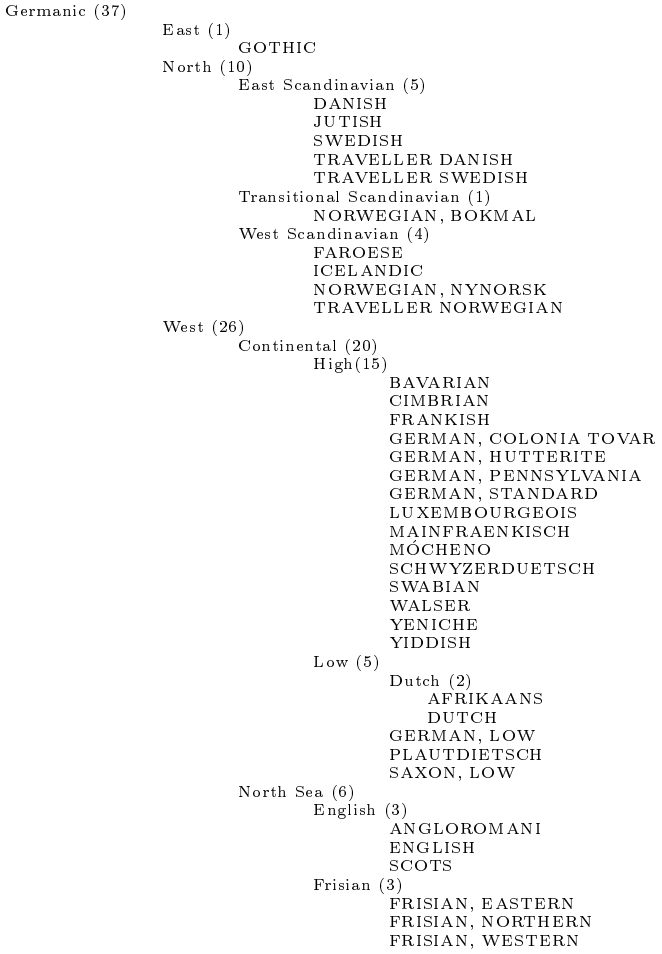


Figure 4.3: Classification of the Germanic language family according to the Ethnologue (1996)

and North Germanic language families, the values of which are respectively 1.71 for West and 1.19 for North. No table has been included for East as the East Germanic language family does not have any intermediate level. Consequently, the DV for East is 0.

D. Choice of languages

Rijkhoff states that, in order to obtain the right variation across language families, we first have to assign one language to each language family which on the basis of its linguistic diversity value alone would not be represented in the sample (in this case, families with a linguistic diversity lower than 2.9/5= 0.58[3]). This means that 1 language has to be taken out of East, whose linguistic diversity equals 0. The remaining 4 languages have to be chosen proportionally with the linguistic diversity within the Germanic subfamilies in order to obtain maximal variation of

[3]This value is the result of the total DV (1.71 + 1.19 + 0) divided by the sample size.

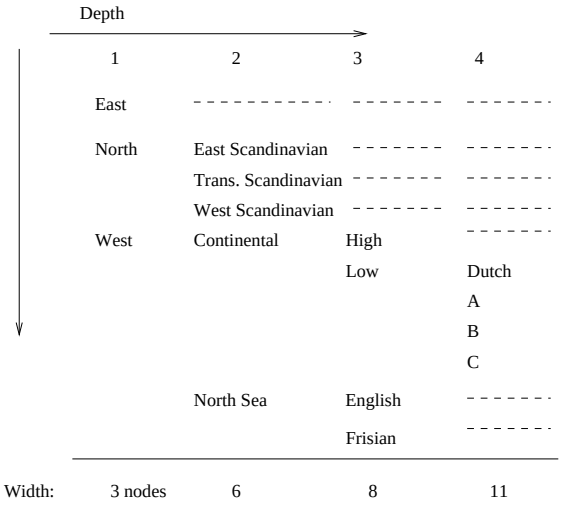


Figure 4.4: Width of the four intermediate levels of the Germanic language family

data. Since the linguistic diversity values of West and North are respectively 1.71 and 1.19, the remaining 4 languages are equally distributed over West and North: two languages from the North Germanic language family, and two from the West Germanic language family.

It is important to note that Rijkhoff's method does not choose the actual languages, but only determines from which groups the languages have to be taken. In general, the actual choice of languages will depend heavily on the availability of adequate language descriptions and data. Also the investigator's knowledge of the subject will play a role. It is more likely that the investigator will choose languages he/she knows. There is also the issue of extinct languages. According to Rijkhoff's method a representative subset of the Germanic language family should include one language out of the East Germanic subfamily. Gothic, however, is an extinct language, and as our primary goal is not to do historical linguistics, we are not particularly interested in including an extinct language in the sample. Completely ignoring the extinct languages, however, does not seem right as they influence the shape of the genetic tree. Therefore, it was decided to take the extinct languages into account during the computation, but not to include them in the resulting sample. This means that the sample size used during computation was 5, but that the actual sample only consists of 4 languages, excluding Gothic. We assume that this will not negatively affect the value of the sample. The sample that we use in the remainder of this thesis consists of Dutch and English from the West Germanic

language family and Danish and Icelandic from the North Germanic language family. If the sample size would be reduced by one, only one language from the North Germanic family would be taken based on the fact that the West Germanic language family is more diverse than the North Germanic language family. This is important as in one of our test sets we ended up including only three languages in full for reasons to be discussed below.

It should be emphasised that a genetic classification has only been used as a reliable means to establish the language sample. There is no claim that the actual structure of the multilingual lexical hierarchy will follow the genetic classification of languages. Genetically related languages do not necessarily belong to the same typological group and vice versa, and the structure of the multilingual hierarchy might therefore depend on the kind of properties shared between the languages. However, rather than developing our own hierarchy, we let ourselves be guided by historical linguistics and use a genetically based language typology in our parameterised lexical sample fragments.

4.2.2 Lexical Sampling

The next step is to select a subset of the vocabulary of the languages of the language sample. For our research purposes – comparing different architectures for multilingual lexical representation – it is important that the test set does not show any bias towards a particular language. For example, if we would look at computer terminology, then our test set would suffer from a bias towards English as computer terminology in the Germanic languages is heavily influenced by English. To avoid this kind of bias, we restricted ourselves to the medical domain. Medical terminology in the Germanic languages is well-established and shares a common origin in Greek and Latin. Consequently, there is no bias towards any of the languages of our sample. We do, however, expect additional uniformity within our sample because of the shared origin of the terms.

The medical vocabulary can be subdivided into three general classes (Wolff, 1984):

1 **General Vocabulary**
This group includes articles, prepositions, conjunctions, quantifiers (i.e. function words) and words or affixes expressing time or measure. Words belonging to this class do not generally carry sublanguage-specific information.

2 **General Vocabulary with sublanguage-specific usage**

To this class belong words of the major lexical classes that, although part of everyday vocabulary, are used in well-defined and much narrower senses in the sublanguage. The noun *foot*, for example, has several meanings in common English (a body part, the bottom (of a mountain, of a page, etc.), a unit of length, and a unit of metrical verse), whereas only a subset of those will occur in medical texts. Frequently, these popular medical terms have a corresponding technical term.

3 **Medical terminology**

This group comprises all words specific to the subject area and these words are generally reserved for the exchange of information among experts in the field. Words of this group are almost exclusively nouns and adjectives and in the Western European languages the group is marked by a prevalence of words of Greek and Latin extraction.

Our test data consists of vocabulary belonging to class 2 and 3. Two test sets[4] were compiled. A small test set was collected consisting of 19 class 2 words, i.e. body part terms, in Dutch, English, Danish, and Icelandic[5]. The phonemic transcriptions were extracted from CELEX (Baayen, Piepenbrock, and van Rijn, 1995) for Dutch and English, from Hansen (1990) for Danish and from Blondäl (1920 1924) and Einarsson (1945) for Icelandic. The list of body part terms is included in Appendix B.1.

A larger sample consisting of vocabulary belonging to group 2 and 3 was taken using the online Multilingual Glossary of technical and popular medical terms in nine European languages – English, French, German, Dutch, Spanish, Portuguese, Italian, Greek and Danish – (1995)[6]. This test set contains Dutch, English, and Danish[7]. The Multilingual Medical Glossary was developed to provide assistance to the authors of patient information leaflets. The authors of patient information

[4]For the sake of simplicity the test sets are based on lists of translation equivalents for selecting corresponding lexical entries across languages.

[5]The Icelandic equivalents for *Hand* and *Mouth* are missing from the test set.

[6]`http://allserv.rug.ac.be/~rvdstich/eugloss/welcome.html`.

[7]Icelandic has not been fully included in this test set because of difficulties in obtaining the necessary data. First the medical glossary does not contain Icelandic. Although most of the Icelandic terms could be acquired through translation using the Icelandic term bank on the web (`http://www.ismal.hi.is/ob/`), obtaining the phonemic transcriptions was another matter due to the lack of resources. The only dictionaries containing phonemic transcriptions for Icelandic at our disposal were Blondäl (1920 1924) and Einarsson (1945), which are not specialised enough and slightly out of date. Therefore they did not contain any of the more recent medical terms.

leaflets are academically trained physicians and pharmacists, who, in general, have not benefited from any specific linguistic training. Consequently they will have to overcome considerable hurdles at the terminological level when attempting to write texts in plain language meant to be read by lay persons. The Multilingual Medical Glossary helps them by providing technical medical terms with their popular equivalents.

The glossary consists of a list of 1830 scientific medical terms which were extracted from a compendium of scientific data sheets on medical products using frequency. The popular equivalents of these scientific terms were obtained through translation. It should be noted that in each of the nine vocabulary lists, there are a number of terms for which either a scientific or a popular equivalent or even both are lacking. A scientific term can be missing in one of the languages because no translation could be found. A popular equivalent can be missing because a popular term does not exist for the scientific term in question – it is impossible to explain the scientific term in understandable lay language – or because of overlap between scientific and popular terms. The latter is more common in the Romance languages than in the Germanic languages. Being closely related to Latin, many scientific medical terms of a Latin origin pose no problem to native speakers of Romance languages, whereas they are not understandable for native speakers of Germanic languages. Note that within the Germanic languages, English shows more cases of overlap between scientific and popular terms than the other Germanic languages because of the relatively large Romance component in its vocabulary.

From the list of 1830 English scientific medical terms, a set of 400 terms was chosen at random. For each of those the corresponding scientific and popular terms were looked up in the other languages. From the resulting list only adjectives and nouns were selected for which a technical medical term and a phonemic transcription could be found in all languages of the test set. The phonemic transcriptions were again extracted from CELEX (Baayen, Piepenbrock, and van Rijn, 1995) for Dutch and English, from Hansen (1990) for Danish. Popular equivalents of the technical terms were only included if they consisted of a single term and were different from the scientific term in all three languages. The final data set consists of 71 nouns and 29 adjectives[8]. In addition, a small test set for Icelandic was compiled consisting of a subset of the medical test set (25 nouns). The Icelandic data was used to

[8]There is one term which is an adjective in Dutch and English but gets translated as a noun in Danish.

test the extensibility of the lexical fragments. The medical test set is included in Appendix B.2.

4.3 Multilingual Information Sharing

The second methodological issue involved in the implementation of the sample fragments is the sharing of information in a multilingual inheritance network. This issue can be split up into three questions a) which information **can** be shared between the languages in the lexicon, b) which information **should** be shared between the languages in the lexicon, and c) **how** can it be shared?

Which information can be shared?

As mentioned earlier, we focus in this thesis on the sharing of morphological, phonological, and morphophonological aspects of word formation and all other aspects of linguistic description are ignored including semantics. This does not mean that we think that semantics is not important, we just do not focus on it. Some semantics is, however, implicitly present in our fragments as translation equivalents were used for selecting corresponding lexical entries across languages. In Chapter 7, we discuss briefly how semantics could be integrated in our multilingual inheritance lexicons.

Since the languages in our test sets are historically related, it is no surprise that they share many characteristics at the morphological, phonological, and morphophonological level. The phonological form of words which come from a single root are more or less similar in the different languages depending on the diversifications that have taken place in each language. Consider, for example, the word *cat* which transcribes as /k{t/ in English, /kAt/ in Dutch, and /kad/ in Danish. Here the value of the onset is shared by all three languages and the value of the coda is shared by two. Danish uses a /d/ rather than a /t/. Even the values of the vowels (/{/, /A/, and /a/) are virtually the same. They have slightly different realisations, but are phonologically non-distinctive, i.e. if the Dutch /A/ were substituted by the English /{/ in Dutch, the result would not be a different word, but it would simply sound like a different accent. Such cross-linguistic phoneme correspondences can be captured by introducing metaphonemes (Tiberius and Cahill, 2000a; Tiberius and Cahill, 2000b). A metaphoneme is a generalisation

over language-specific phonemes of which the realisation at the multilingual level is determined by the choice of language. Thus, to capture the above mentioned /{ - A - a/ correspondence, we could introduce a metaphoneme |{Aa| which is realised as an /{/ in English, an /A/ in Dutch, and an /a/ in Danish. In the context of this thesis, metaphonemes were defined and incorporated in the sample fragments for the vowel phonemes in Dutch, English, and Danish[9].

An example of a morphological property which is shared by two of the languages of the test set is the plural noun ending in -s which occurs in Dutch and English, but not in Danish or Icelandic. For example, *car* in English and *auto* 'car' in Dutch both have a plural ending in -s. In English, this plural noun suffix has distinct phonological realisations determined by phonological context. If the base ends in a sibilant, it is realised as /Iz/, if the base ends in a vowel or a voiced consonant other than a sibilant, it is realised as /z/, and if the base ends in a voiceless consonant other than a sibilant, it is realised as /s/. In Dutch, no morphophonological changes occur.

The morphophonology of the Germanic languages also exhibits a range of similarities. A common morphophonological feature of the Germanic languages is, for example, a vowel change which occurs in the plural form of a subset of nouns. Compare:

	singular			**plural**	
English	*foot*		/fUt/	*feet*	/fi:t/
Dutch	*stad*	'town'	/stAt/	*steden*	/ste:-d@/
Danish	*mand*	'man'	/man/	*mænd*	/mEn/
German	*Man*	'man'	/man/	*Männer*	/mE-n@r/

Which information should be shared?

There are in principle four reasons why languages may exhibit similarities (Comrie, 1989). First, resemblances can arise due to chance. An example of this is German *nass* and Zuni (an American Indian Language of New Mexico) *nas* which both mean *wet* (Ruhlen, 1987). Second, two languages may exhibit similarities because of language contact and one language could have borrowed a property or a lexical

[9]The results can be found on http://www.itri.brighton.ac.uk/~Carole.Tiberius/mphon.html. A cross-linguistic study of consonant phoneme correspondences is undertaken by Lynne Cahill under ESRC grant N^o R000223681.

item from the other. Lexical borrowings generally affect certain semantic domains such as previously unknown cultural items, e.g. *kangaroo*, *tobacco*, and certain grammatical categories, e.g. nouns rather than verbs (Ruhlen, 1987). Third, languages may exhibit similarities because they are genetically related and have inherited a common property from a common ancestor. This was illustrated above with the phonological form of *cat*. Fourth, a shared property could be a language universal, either absolute or a tendency. For example, word order.

As we saw in Chapter 2, some projects make a distinction between these different kinds of similarities. The KPML project, for example, is primarily concerned with capturing near-universals at the syntactic level ('almost all languages can make sensible use of a declarative/interrogative distinction'). In PolyLex and Kameyama's grammar, on the other hand, no distinction is made between the different kinds of similarities and any shared information is captured whether it is due to typology, genetics, language contact or chance. For example, in PolyLex we will find generalisations such as the default onset for a particular root is a sibilant. In this thesis, we follow the second approach and aim to encode any shared information in the lexical sample fragments. A potential danger of this approach is that if one aims to develop a core lexicon that can be used for further lexical development in related languages, substantial encoding of non-universals could skew the core lexicon inappropriately.

How should the information be shared?

The approach to information sharing adopted in this thesis is mainly **data-driven**[10]. By data-driven we mean that known facts about the languages in the lexicon serve as input for the construction of a multilingual inheritance hierarchy and information which is common to most of the languages in the data is shared. This means that in a first stage all possible attribute-value pairs in the language-specific lexicons are evaluated and in a second stage the most common attribute-value pairs are then shared between the languages in the multilingual lexicon. It is possible that information is shared by only a subset of the languages in the sample. This can be captured by introducing subhierarchies in the multilingual lexical hierarchy.

[10]Data-driven information sharing could be contrasted with theory-driven information sharing (which could be, for example, sharing on the basis of what corresponds to the unmarked case). However, as we do not have a coherent notion of what theory-driven information sharing in a multilingual context involves, this concept has not been pursued in this thesis.

The data-driven approach to information sharing is illustrated in Figure 4.3 for the phonological structure of the lexical entry *Cat* in Dutch, English and Danish which we discussed earlier in this section. All values are shared except for the final coda in Danish.

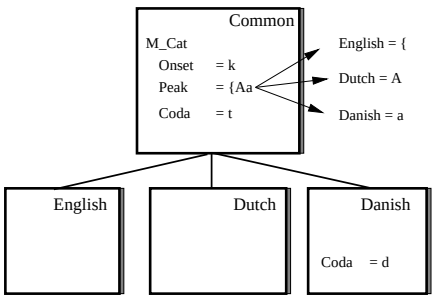


Figure 4.5: Lexical entry for *Cat* in a data-driven approach

Data-driven approaches to multilingual information sharing have also been adopted by Cahill and Gazdar in the PolyLex project (1999b) and by Hippisley and Gazdar (2000) to construct a hierarchy of colour terms in the Slavonic languages.

4.4 Development Strategy

The third methodological issue that we consider concerns development strategies. There are several ways one can go about developing a multilingual hierarchical lexicon which captures linguistic similarities between related languages.

First, a **parallel** development strategy can be adopted. This means that the lexicons for the different languages are developed in parallel and that cross-linguistic generalisations are captured immediately upon construction. Parallel development implies an ideal situation in which it is possible to do a thorough analysis of all the languages in the lexicon prior to implementation in order to get an overview of the linguistic phenomena that the lexicon should be able to describe. If this is not possible, it is unlikely that optimal sharing can be achieved using this development strategy.

Second, it is possible to use a **non-parallel** development strategy. Broadly, two variants of non-parallel development can be distinguished:

- All the monolingual lexicons are first fully developed separately before they

are integrated into a multilingual lexicon capturing the similarities that exist between them. This development strategy favours optimal sharing.

- The lexicons for the different languages are developed and integrated successively. First, the lexicon for one of the languages is fully developed, then the lexicon for a second language will be developed in terms of the first lexicon, etc. A common strategy in this case is to develop the lexicon of the richest language first and then to add the other lexicons as more impoverished versions of the first lexicon. This strategy was adopted by Beard (1981) in his development of a lexical theory for the Indo-European languages.

Rather than adopting a parallel or non-parallel development strategy, it is also possible to combine the two, moving for instance from a non-parallel development strategy in the early stages to a parallel development at a later stage (and vice versa). Both parallel and non-parallel development strategies are explored in this thesis.

4.5 Evaluation

This section gives an overview of measures that will be used to evaluate the multilingual inheritance architectures defined in Chapter 3. It is inspired by related work on the evaluation of inheritance networks in machine learning in DATR (Barg, 1994; Barg, 1996; Lock, 2000). To evaluate the multilingual architectures, the validity and quality of the implemented sample fragments will be determined.

The **validity** of a sample fragment can be established by determining whether it is consistent and complete with respect to the input data. For us a fragment is consistent if it provides a description of the facts as they are given and it is complete if it covers all input data. These criteria are taken from Barg (1994; 1996), but their definition has slightly been altered. Barg regards consistency and completeness as minimal requirements that a semantic network must satisfy to claim that it covers a given set of observations.

Once the validity of the sample fragments has been determined, we need to judge their **quality** to see which sample fragment gives the best DATR theory. Lock (2000) uses amongst others, the size of the DATR theory to determine its quality. She discusses various ways in which the size of a DATR theory can be measured. The

size can be measured by counting the absolute or average number of sentences per object, an object being a node. The hierarchy with the lowest number is the best. Another way to measure the size of the resulting hierarchy is to count the number of *attribute* literals. This corresponds to the number of features mentioned in the hierarchy. We will use the average number of sentences per object to measure the size of the sample fragments.

Another criterion that will be used to determine the quality of the sample fragments is inferential complexity. Inferential complexity can be measured by counting the number of steps that is required to look up a particular value in the sample fragment.

In addition we evaluate the quality of the overall models by determining the flexibility of the models with respect to development strategies and by establishing their robustness and extensibility. For each model, we discuss whether both a parallel and a non-parallel development strategy can be used or only one of the two? In order to evaluate the extensibility of a model, we determine how easy it is to add the Icelandic data set to the sample fragments of the medical test set in the different models.

4.6 Summary

This chapter discussed methodological issues involved in the implementation of the sample lexicons that we use to compare the different architectures for multilingual lexical representation described in Chapter 3. The test data was selected, the approach to information sharing was defined, different development strategies were discussed, and evaluation measures were established. Before we turn to the actual implementation in Chapter 6, we first discuss the lexical description framework that has been used in the sample fragments in Chapter 5.

Chapter 5

The Lexical Description Framework

5.1 Introduction

The lexical description framework[1] that is used in the sample fragments focuses on the representation of morphology, phonology and morphophonology. Using a highly modular default inheritance-based approach, it supports the description of lexical generalisations traditionally modelled as morphology and phonology in a single phonological feature based representation. The framework is perhaps most straightforwardly viewed as a development of the PolyLex lexical description framework (Cahill and Gazdar, 1997; Cahill and Gazdar, 1999a; Cahill and Gazdar, 1999b), extending the PolyLex word model down to the level of phonological features and adopting a more modular and more uniform phonologically-based approach to lexical generalisation. This way it provides a more flexible means of capturing lexical generalisations within and across languages. Section 5.2 describes the framework based on Tiberius and Evans (2000). The framework is implemented in DATR (Evans and Gazdar, 1996). Section 5.3 discusses some of DATR's more advanced features that are used by the framework. Finally, the

[1]The lexical description framework discussed in this chapter is based on joint work by Roger Evans and Carola Tiberius and draws on material presented as Tiberius and Evans (2000). A copy of this paper is included in Appendix ??.
Roger Evans was responsible for the development of the abstract framework and the implementation of the framework in DATR.
Carola Tiberius was responsible for the linguistic design and implementation of the framework including aspects such as data collection, testing, and evaluation of the framework.

framework is illustrated with a sample implementation in Section 5.4.

5.2 The lexical description framework

5.2.1 Theoretical Background

Our framework draws heavily on the theoretical work from Cahill (1990) which has been further developed in Cahill (1993) and Cahill and Gazdar (1997; 1999a). The idea behind Cahill's *syllable-based morphology* is that since many morphological alternations are phonologically based, they can be best described as mappings between sequences of tree-structured syllables. Effectively, morphological operations are defined in terms of changes to the phonology. For example, German umlaut (Apfel – Äpfel) will be represented as a change of the vowel (peak) of the first syllable. Gibbon (1992) adopts a very similar position, although his work is more tuned towards lexicons for speech applications integrating phonological information above the level of the syllable, such as metrical structure.

Both these approaches still make a distinction between morphology and phonology. However, they do not adopt the traditional notion of level of description, or of rules mapping from one level to the other. For them, the linguistic description is just a set of simultaneously applicable constraints. These constraints may, for example, directly connect morphosyntactic attributes to individual phonological components of word forms.

Our framework pushes this view further making no sharp distinction between morphology and phonology at all. We start with the actual structures that we are aiming to describe, and generalise over them motivated purely by structural considerations, without any preconceptions about whether the generalisations are phonological or morphological. The initial structures are phonological, and so the generalisations are phonological. There are echoes of traditional morphology, for example generalisations which correspond to some extent to traditional 'morphemes', but no explicit separation or reference to morphology is required.

Nevertheless, our approach does reveal **structural** distinctions which induce a high degree of modularity in the representations. Like Zwicky (1990), our framework opts for something like the subcomponent divisions of traditional grammar, rather

than the level or strata of 'lexical morphology/phonology'. The following components are distinguished – lexemes, syllable sequences, syllables and phonemes.

Following Cahill and Gazdar (1997; 1999a), a segmental model of phonology is adopted in which phonological units are discrete and in simple temporal sequence. However, rather than using phonemic transcriptions, where the primitives are vowels and consonants, our framework goes down to the level of phonological features. This permits a more accurate and a more elegant treatment of phenomena such as elision, final consonant devoicing, vowel lengthening, and assimilation (e.g. Cahill, 1993; Coleman, 1992; Bird and Klein, 1990). For example, vowel lengthening involves just a change in the length feature of the vowel, regardless of the particular vowel involved, whilst final consonant devoicing just changes the voice feature of final consonants[2].

The framework concentrates on the treatment of inflection. The general approach is of what Stump calls the **inferential-realisational** type (e.g. Zwicky, 1985; 1990; Anderson, 1988; Stump, 2001). In theories of this type, paradigms (inflectional classes, declensions, conjugations, etc.) are treated as analytically central, rather than epiphenomenal or of secondary status. The central notion in these theories is the lexeme, not the word or the morpheme. Words exist as realisations of morphosyntactic specifications of lexemes: an inflected word's association with a particular set of morphosyntactic properties licenses the application of rules determining the word's inflectional form. For example, the English word *likes* arises by means of a rule appending *-s* to the stem *like* which has the properties 'third-person singular subject agreement', 'present tense', and 'indicative mood'. In the framework, lexemes, represented as DATR nodes, are the primary content of a lexicon and word forms are accessed by applying lexical operations (implemented using the lexical rule techniques described in Evans, Gazdar and Weir (1995; 2000) and Smets and Evans (1998)) such as `singular`, `definite` or `third-person` to lexemes.

The framework makes extensive use of default inheritance to capture linguistic generalisations. In this sense it is closely related to Corbett and Fraser's Network Morphology (Corbett and Fraser, 1993), which treats language as a network of interacting parallel hierarchies of linguistic knowledge.

[2]Of course, a segmental description is still an idealisation of reality – see Cahill, Carson-Berndsen, and Gazdar (2000, p.97) for a discussion of how a segmental description can be extended to deal with nonsegmental issues.

5.2.2 Objects of description

The primary objects the lexicon aims to describe are word forms, or more precisely phonological analyses of word forms. Such word forms are viewed as labelled tree structures with a root representing the whole word form and successive decomposition into syllable sequences and then syllables. Each syllable has the conventional structure shown in Figure 5.1, which was originally proposed by Pike and Pike (1947). Thus a syllable consists of an onset (the initial consonant cluster)

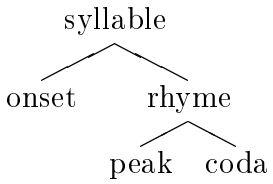


Figure 5.1: Conventional syllable structure

and a rhyme, and the rhyme consists of a peak (the vowel(s)) and a coda (the final consonant cluster).

An example of the tree structure associated with the singular of the lexeme *Hand* in English, is given in Figure 5.2. Each vowel or consonant phoneme is represented by a full feature set: for example, the first phoneme of the onset of *Hand* is a voiceless fricative glottal consonant[3]. Where sequences of components (syllables, phonemes etc.) occur, they are numbered from left to right. In the case of *Hand*, the syllable sequence contains a single syllable, labelled `1`, and the onset and peak of that syllable contain single phonemes, but in the coda we have two phonemes labelled `1` and `2`. In our model, multiple peaks are only used to describe diphthongs. Long vowels are considered as single peaks[4].

In Figure 5.2, the analysis consisted of a single root represented as a (1 element) syllable sequence. For technical convenience, more complex word forms are described using a **concatenation** node `concat`. Such concatenations are viewed themselves as syllable sequences (but not flat ones), which means that the representation space is recursively defined: `concat` can dominate flat syllable sequences

[3]The feature classification that has been used is based on Cahill, Carson-Berndsen and Gazdar (2000) We have extended this classification with one extra tier called `syllabicity` to account for syllabic consonants and non-syllabic vowels, as for example the syllabic `/l/` in the English word *ankle*.

[4]Whether diphthongs and long vowels should be treated differently is still a matter of debate.

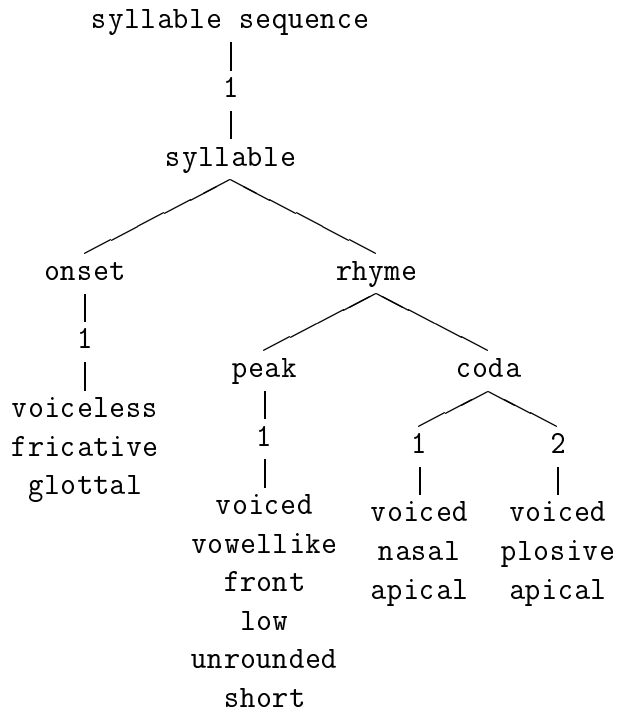


Figure 5.2: Wordform structure for *Hand*

(primitive units of word formation) or other `concat` nodes, allowing more complex word forms to be represented. Note however that no distinction is made between prefixation and suffixation, or roots and affixes: `concat` is simple left-right concatenation. Figure 5.3 shows the syllable sequence structure for the English plural form *fingers*, consisting of the concatenation of a two element sequence *finger* and the single element sequence s[5]. Notice that this last element would not conventionally be considered a complete phonological syllable, but at this level of the analysis it is treated in the same way as other 'real' syllables. We do not address resyllabification issues[6].

5.2.3 The organisational structure of the lexicon

This section outlines the way default inheritance and rule application are used to represent word forms, as discussed in the previous section, compactly in the

[5]Here and below SAM-PA CELEX transcriptions (Baayen, Piepenbrock, and van Rijn, 1995; Wells, 1987; Wells, 1989) are used to abbreviate actual phonological feature bundles where the feature details are not important.

[6]Resyllabification goes across word boundaries and must therefore be dealt with above the word level.

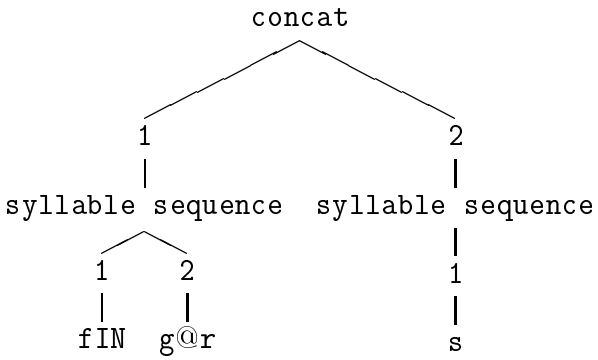


Figure 5.3: Word form structure for *fingers*

lexicon.

The framework defines word form trees **equationally**. That is, they are described by sets of equations each of which associates a path from root to leaf in a tree with one of the phonological feature values specified at that leaf. For example, the `Onset` subtree in Figure 5.2 can be described by the following three equations:

```
phon syll 1 onset 1 phonation = voiceless
phon syll 1 onset 1 manner = fricative
phon syll 1 onset 1 place = glottal
```

These equations are not simply listed for each word form, however, but are organised into an inheritance hierarchy. There are three main components to the organisational structure of the lexicon.

First, the framework is **lexeme-based**. Each lexeme defines a base tree, which need not correspond to any actual word form. Individual word forms are obtained by applying morphosyntactic functions such as `singular, plural, present, past, nominal, gerund` to a lexeme. Functions can also be combined to produce further word forms. For example, applying `nominal+plural` to the lexeme for *Love* gives the structure for the word form *lovers*.

Second, the lexeme definitions are represented using **default inheritance**. A typical lexeme needs to specify explicitly its own basic phonological structure, but can generally inherit information on how it forms a plural, or nominal, or genitive etc. from more abstract classes. If it happens to have, say, an irregular plural, it will specify this in the lexeme entry, overriding just that part of the inherited information.

Third, the internal components that represent the phonological form of a lexeme are organised into their own independent inheritance hierarchies – the syllable sequence hierarchy, the syllable hierarchy, and the phoneme hierarchy. The lexeme hierarchy inherits the phonological form of a word form from the syllable sequence hierarchy, which inherits individual syllable structures from the syllable hierarchy, which inherits individual phonemes from the phoneme hierarchy. Figure 5.4 illustrates this for the Dutch lexeme *Gebed* ('prayer'). As a lexeme, *Gebed* is primarily linked into the lexeme hierarchy, inheriting from `Noun_EN`, a subclass of `Noun`. But it inherits part of its content, namely its phonological form, from `GEBED` in the syllable sequence hierarchy. `GEBED` is primarily a `Disyllable`, but it inherits part of its content, namely the two syllables it contains, from `GE` and `BED` in the syllable hierarchy. Finally the syllable `BED` inherits part of its structure, from the consonants `b` and `d` and the vowel `E` in the phoneme hierarchy.

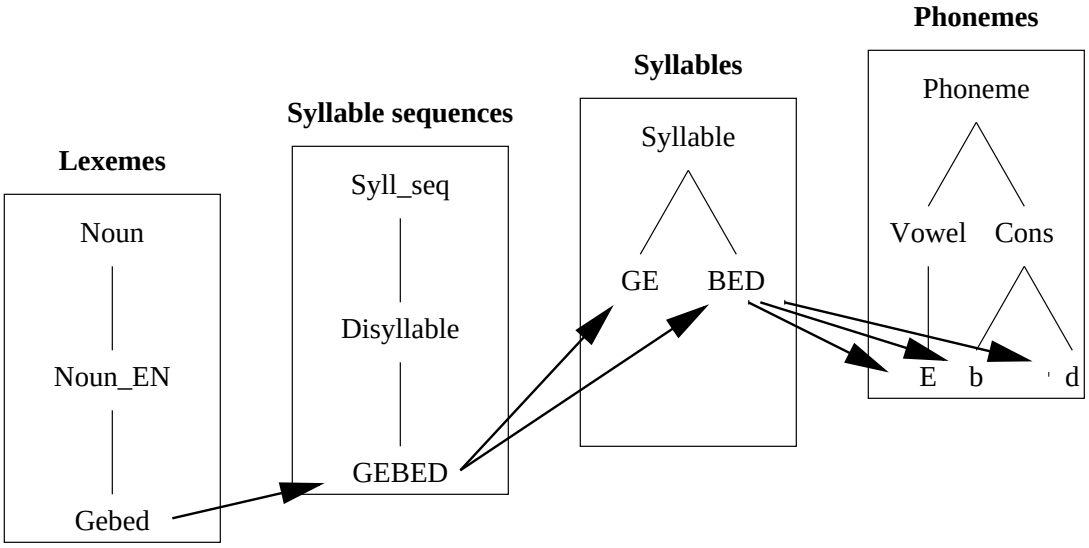


Figure 5.4: Module and node structure for lexeme *Gebed*

The lexeme access functions are implemented using the **lexical rule** techniques described in Evans, Gazdar and Weir (1995; 2000), Smets and Evans (1998). Each rule maps an input word form structure onto an output word form structure. Sometimes rules are chained together to obtain a particular word form. The first rule applies to the lexeme input, the second to the output of the first and so on – the final word form is the output of the last rule in the chain.

Lexical rules are specified in the same inheritance structures as the other tree equations in our framework. This means that the inheritance hierarchy can be

used to control the scope of applicability of a rule: a universal rule can be defined at the top-most node of the hierarchy, one which only applies to nouns at the `Noun` node, one that applies to verbs at the `Verb` node etc. Individual words can even have their own rules if required. In addition, rule definitions can inherit and override from ancestor nodes just as lexeme definitions can: the `Noun_EN` node inherits the `plural` rule from `Noun`, retaining the fact that pluralisation is achieved by concatenating something to the root, but overriding what is concatenated – *-en* instead of *-s*, if *-s* is the default plural ending.

Rules may include conditional constructs, testing properties of their input word form structures to decide whether or how to apply. This can be used to control the scope of rule application, for example blocking `superlative` on adjectives that are already in the comparative form, or to control the effect of rule application, for example choosing between `/s/`, `/z/`, and `/Iz/` as a plural noun suffix in English. More fundamentally, it is also often used to control where in a word form the rule applies. Because word forms are defined equationally, each rule operates on every equation of the word form definition. For most of these it will do nothing, being activated only, for example, in equations relating to the last syllable, or the first peak vowel.

In our framework, rules can be defined in each of the submodule hierarchies (as shown in Figure 5.4) independently. Rules in the lexeme hierarchy are invoked directly on lexical access (e.g. `singular`), but other rules can be invoked only from the hierarchy above them. This will be illustrated with the rule for final devoicing in Section 5.4 below.

The implementation of the lexical rule mechanism will be discussed in the next section which introduces some of DATR's more advanced features that are used by the framework.

5.3 Implementation of the framework in DATR

Parts of our lexical description framework exploit some of DATR's more advanced features[7]. This concerns in particular the use of lexical rules.

[7] For a more detailed description of DATR the reader is referred to Evans and Gazdar (1996) and the DATR web pages `http://www.datr.org`.

Lexical Rules

In our framework, lexical rules are inserted before the 'ordinary' object part of the DATR path expression. Thus path expressions now consist of two parts: a rule part and an object part. The structure of the object part was illustrated above. The rule part contains a list of zero or more rule names. In order for a sequence of rules to have effect, the various input-output paths have to be linked together using inheritance, creating a chain of inheritances between the base and the derived tree structures of the lexical entry. This is achieved by the following code[8]:

```
RULE:
    <$rule1 $rule2> == <$rule2>
    <$rule> == Local:<GlobalNode rule $rule>.

INPUT:
    <> == "< "PREVRULE" >".

PREVRULE:
    <$rule1 $rule2> == $rule1 <$rule2>
    <$rule> == .
```

Rule application works like backward chaining. For example, if we have a rule sequence consisting of two rules <`rule1 rule2`>, then the rule mechanism will first get the last rule in the sequence (first equation of `RULE`) and try to apply it. In this case, this is `rule2`. By default, the output of a rule is the same as the input, so that lexical relationships need only concern themselves with components they modify. Thus, `rule2` looks for its input and finds that it needs the output of some previous rule as its input. `PREVRULE` returns the name of this rule, `rule1`. `INPUT` brings us back to the global context, where `RULE` gets invoked again. `RULE` applies now rule `rule1` which takes the base form as its input. The output of `rule1` is then used by `rule2` as its input to form the final output. This is illustrated in Figure 5.5. An example of rule chaining with real data is given in Section 5.4 with

[8] $rule, $rule1, $rule2 are variables which will be instantiated to the rule names whose input we are seeking. The nodes `GlobalNode` and `Local` are DATR-2.8 additions. This also counts for the nodes `Global`, `QueryNode`, and `Query` which are used later. They are part of the DATR Standard Library RFC. `Local` enables one to create a local descriptor from the names of its parts. Thus `Local:<Node Path>` sets the local context to `Node:Path`. `GlobalNode` returns the name of the global node. For more information on these functions, see the DATR web pages.

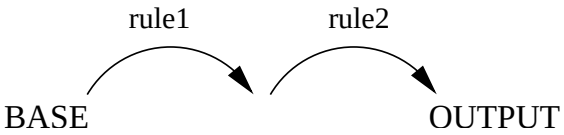


Figure 5.5: Example of rule chaining

the definition of the definite plural form of Danish nouns.

Special Nodes

Figure 5.4 illustrated how the information that defines a particular lexeme is specified across the different submodules of our framework. Each of these modules forms its own global context and when we go from one module to the next, e.g. from syllable sequence to syllable, the global context changes. The global context is used to check the state of the current module. The Query context, on the other hand, stays the same for the whole analysis and is used to check the original query. Two nodes, `G_NEWOBJ` and `Q_NEWOBJ`, were defined which help us to do this:

```
G_NEWOBJ:
    <> == Global:<GlobalNode RULEPART:<GlobalPath>>.
```

The node `G_NEWOBJ` takes an object path, evaluates it as if it was the object path of the global context, attaches the original rule part to it and does a global inheritance.

```
Q_NEWOBJ:
    <> == Query:<QueryNode RULEPART:<QueryPath>>.
```

The node `Q_NEWOBJ` takes an object path, evaluates it as if it was the object path of the original query, attaches the original rule part to it and does a query level inheritance.

Display Routines

The implementation of our framework also includes routines which allow us to bundle up collections of what we call macro features at the object level to return the whole analysis represented by a dag as a single value. In general, these macro features traverse the dag and concatenate what they find at the leaves. Two macro

features are distinguished <macro segmental> which returns the entire phonological form in a segmental phonemic representation and <macro featural> which returns a feature-based representation. For example, the English lexeme *Arm* is described by the following tree equations:

```
phon syll 1 rhyme peak 1 syllabicity = [syllabic]
phon syll 1 rhyme peak 1 phonation = [voiced]
phon syll 1 rhyme peak 1 manner = [vowellike]
phon syll 1 rhyme peak 1 place = [back]
phon syll 1 rhyme peak 1 height = [low]
phon syll 1 rhyme peak 1 roundness = [unrounded]
phon syll 1 rhyme peak 1 length = [long]
phon syll 1 rhyme coda 1 syllabicity = [nonsyllabic]
phon syll 1 rhyme coda 1 phonation = [voiced]
phon syll 1 rhyme coda 1 manner = [nasal]
phon syll 1 rhyme coda 1 place = [labial]
```

The macro features group this information together and return the following:

```
Arm:<macro segmental> = { A: m }.
Arm:<macro featural> = [ { [ [syllabic] [voiced] [vowellike] [back]
 [low] [unrounded] [long] ] [ [nonsyllabic] [voiced] [nasal]
 [labial] ] } ].
```

The macro features are really a diagnostic aid, and not part of the description proper.

All this machinery forms the DATR infrastructure of our lexical description framework. It is defined in a separate file and is used as such in the sample implementations of each model.

5.4 Illustration of the framework

This section discusses a sample implementation in DATR to demonstrate the key features of our lexical description framework. For a larger sample fragment, the reader is referred to the appendices which include the implementation of the body

part sample lexicons. First, the implementation of the different submodules – lexeme, syllable sequence, syllable, and phoneme (see Figure 5.4) – will be described. Then, the lexical rule application will be illustrated with a few example rules.

The first module, the **lexeme** module, specifies the main inflectional inheritance hierarchy. Here we focus on Dutch noun inflection and a fragment of a lexical hierarchy of Dutch nouns is shown in Figure 5.6. This hierarchy describes the

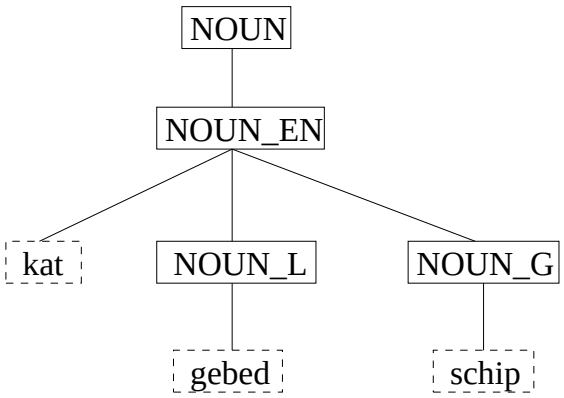


Figure 5.6: Fragment of a lexical hierarchy of Dutch noun inflection

subset of Dutch nouns which have a plural ending in *-en*. Two subclasses can be distinguished within this noun class, i.e. `NOUN_L` and `NOUN_G`. `NOUN_L` contains those nouns which in addition to the ending *-en* get vowel lengthening in plural, e.g. *gebed* `/x@bEt/` - *gebeden* `/x@be:d@/` ('prayer'). `NOUN_G` describes nouns with *-en* ending and vowel gradation, e.g. *schip* `/sxIp/` - *schepen* `/sxe:p@/` ('ship'). These classes are lexically conditioned.

In DATR, this lexical hierarchy can be implemented as follows:

```
NOUN:
    <> == CATEGORY
    <syn cat> == noun
    <number> == sing
    <rule singular> == SINGULAR_NOUN_RULE:<>.

NOUN_EN:
    <> == NOUN
    <rule plural> == PLURAL_NOUN_EN_RULE:<>.
```

```
NOUN_L:
    <> == NOUN_EN
    <rule plural> == PLURAL_NOUN_L_RULE:<>.

NOUN_G:
    <> == NOUN_EN
    <rule plural> == PLURAL_NOUN_G_RULE:<>.
```

As mentioned earlier, the lexeme representations themselves do not have word forms associated with them. They are abstract forms from which actual word forms are constructed through inflectional and derivational processes. Actual word forms are obtained by specifying functions corresponding to lexical notions such as `singular, plural`, etc. Thus to obtain the singular form, the rule `singular` is applied, to obtain plural, the rule `plural` is applied. Note here that the content of the rules is not defined directly at the node where it is invoked, but an auxiliary node is introduced. This is done for convenience. It makes the code neater and easier to read. For example, the rule for `singular` is defined as follows:

```
SINGULAR_NOUN_RULE:
    <> == INPUT
    <phon> == INPUT:<phon devoicelastsylllastcoda>.
```

This rule takes an input base tree defined in a lexical entry and applies final devoicing to it, as Dutch singular nouns are subject to final devoicing. The final devoicing rule for Dutch will be discussed later in this section.

Lexemes inherit from one of the noun classes in the lexical hierarchy. For example, the lexeme *Kat* 'cat' inherits from `NOUN_EN` and its definition looks like this:

```
Kat:
    <> == NOUN_EN
    <phon> == "X_KAT:<>".
```

The phonological form for the lexeme *Kat* is not actually specified here, but it is inherited from the other modules – syllable sequence, syllable, and phoneme modules. First, *Kat* will inherit its syllable sequence information from the **syllable sequence** module which is defined as follows:

```
SYLLSEQUENCE:
    <> == RULE
    <type> == syllsequence
    <syll subtype> == 0
    <focus> == "<syll subtype>".

MONOSYLLABLE:
    <> == SYLLSEQUENCE
    <syll subtype> == 1.

DISYLLABLE:
    <> == MONOSYLLABLE
    <syll subtype> == 2.

TRISYLLABLE:
    <> == DISYLLABLE
    <syll subtype> == 3.
```

This fragment supports mono-, di-, and trisyllables, but can easily be extended to cover syllable sequences consisting of four, five, or more syllables.

The feature `focus` is used to refer to the syllable which is likely to undergo morphophonological changes if there are any. By default, the focussed syllable is the final syllable, given by the syllable subtype specification[9].

In the implementation, a distinction is made between the features `type` and `subtype`. The feature `type` is used to refer to the particular modules that are distinguished, and `subtype` is used to refer to the number of components within a module. Thus, the `type` in the syllable sequence module is `syllsequence` and `subtype` is `1` for a monosyllable, `2` for a disyllable, etc.

Kat is a monosyllable and its syllable sequence information is represented as follows:

```
X_KAT:
    <> == MONOSYLLABLE
    <syll 1> == "S_kAt:<>".
```

[9]This reflects the fact that Germanic morphology primarily involves suffixation (cf. PolyLex).

The individual syllable structure is then inherited from the **syllable** hierarchy. The basic syllable is defined as follows:

```
SYLLABLE:
    <> == RULE
    <type> == syllable
    <onset subtype> == 0
    <rhyme peak subtype> == 0
    <rhyme coda subtype> == 0.
```

The feature `subtype` in the last three statements determines the actual number of elements in the onset, peak, and coda of a particular syllable and defaults to none. It gets instantiated in a particular syllable skeleton. Examples of syllable skeletons are:

```
Syllable_V:
    <> == SYLLABLE
    <rhyme peak subtype> == 1.

Syllable_VV:
    <> == SYLLABLE
    <rhyme peak subtype> == 2.

Syllable_CV:
    <> == Syllable_V
    <onset subtype> == 1.

Syllable_CVC:
    <> == Syllable_CV
    <rhyme coda subtype> == 1.

Syllable_CCVC:
    <> == Syllable_CVC
    <onset subtype> == 2.
```

The syllable `S_kAt` for *Kat* is a `Syllable_CVC`. It consists of one element in the onset, one element in the peak, and one element in the coda. The definition of `S_kAt` is as follows:

```
S_kAt:
    <> == Syllable_CVC
    <onset 1> == "C_k:<>"
    <rhyme peak 1> == "V_A:<>"
    <rhyme coda 1> == "C_t:<>".
```

This syllable inherits its individual phonemes from the **phoneme** space. The following code fragment specifies the default feature templates for vowel and consonant phonemes in the phoneme space.

```
Phoneme:
    <> == RULE
    <type> == phoneme.

V:
    <> == Phoneme
    <type> == vowel
    <phonation> == [voiced]
    <manner>    == [vowellike]
    <place>     == [central]
    <height>    == [midlow]
    <roundness> == [unrounded]
    <length>    == [lax].


C:
    <> == Phoneme
    <type> == consonant
    <phonation> == [voiced]
    <manner>    == [fricative]
    <place>     == [apical].
```

In our sample fragments, the neutral vowel schwa `/@/` is assumed to be the default vowel and `/z/` the default consonant. The definition of the vowel `/A/`, for example, looks like this:

```
V_A:
   <>          == V
   <place>     == [back]
   <height>    == [low].
```

With all the (sub)modules in our framework being defined, we can now query our sample fragment, for example, for the value of the feature `height` of the peak in the lexeme *Kat*. The DATR output trace for this query looks like this:

```
== Kat:<phon syll 1 rhyme peak 1 height> ?
** Call :  Kat:<phon syll 1 rhyme peak 1 height>?
** Call :  KAT:<syll 1 rhyme peak 1 height>?
** Call :  S_kAt:<rhyme peak 1 height>?
** Call :  V_A:<height>?
** Exit :  V_A:<height> = [low]?
Kat:<phon syll 1 rhyme peak 1 height> = [low].
```

In this example, no lexical rules were involved. The DATR code and output traces get more complex when lexical rules are invoked. In the remainder of this section, we illustrate lexical rule application with examples of noun inflection in Dutch, English, and Danish.

Our first example concerns the final devoicing rule which we mentioned earlier for Dutch. In Dutch, final consonant devoicing applies to root final obstruents (plosives and fricatives) when the root is not inflected or when an inflectional suffix is added which does not begin with a vowel. In our sample fragments, this is achieved by means of a lexical rule which ultimately sets the `phonation` feature of the last coda to `voiceless`.

The devoicing rule in the lexeme module invokes `devoicelastsylllastcoda` in the syllable sequence module. This does nothing except in the last syllable of the word form, in which it invokes `devoicelastcoda` in the syllable module. This also does nothing except in the last coda of the syllable, in which it invokes `devoice` in the phoneme module. Here it changes the value for `phonation` from `voiced` to `voiceless`. This process is schematically represented below.

Notice that rule definitions at each level are not context dependent: `devoicelastcoda` can devoice the last coda of any syllable, not just the last one, and `devoice` can devoice any phoneme.

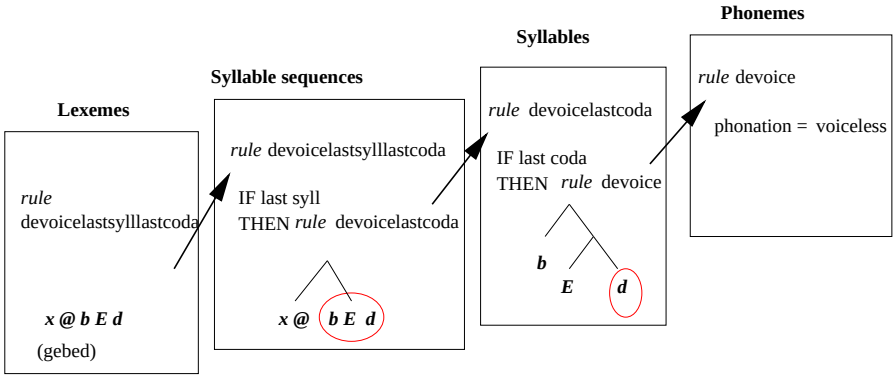


Figure 5.7: Final devoicing applied to the Dutch lexeme *Gebed* ('prayer')

Let us now consider the definition of final devoicing in DATR. We saw in the definition of `SINGULAR_NOUN_RULE` in Dutch how the devoicing rule starts at the lexeme level by invoking `devoicelastsylllastcoda` in the syllable sequence module. For ease of reference, the definition of `SINGULAR_NOUN_RULE` is repeated here.

```
SINGULAR_NOUN_RULE:
    <> == INPUT
    <phon> == INPUT:<phon devoicelastsylllastcoda>.
```

This rule states that the output form is the same as the input except for devoicing of the last coda of the last syllable of the root. The rule `devoicelastsylllastcoda` is invoked in the syllable sequence space and looks for the last syllable of the root and then applies `devoicelastcoda` to it. To this end, the following statement has to be added to the `SYLLSEQUENCE` node:

```
SYLLSEQUENCE:
    ...
    <rule devoicelastsylllastcoda> == DEVOICELASTSYLLLASTCODA:<>.
```

The content of `DEVOICELASTSYLLLASTCODA` is defined as:

```
DEVOICELASTSYLLLASTCODA:
    <> == FINDLASTSYLLROOT
    <syllrule> == devoicelastcoda.
```

This rule finds the last syllable of the root and then applies the rule devoicelastcoda to that syllable. To this end, the following rule has to be added to the SYLLABLE node:

```
SYLLABLE:
    ...
    <rule devoicelastcoda> == DEVOICELASTCODA:<>.
```

The content of DEVOICELASTCODA is defined as follows:

```
DEVOICELASTCODA:
     <> == INPUT
     <rhyme coda> == FINDLASTCODA:<>
     <codarule> == devoice.
```

Nodes such as FINDLASTSYLLLASTROOT and FINDLASTCODA can be considered to be built-in functions which allow us to look for a particular part in the syllable (sequence) structure and to apply a rule to it.

Once the last coda is found, devoice sets the phonation feature of this phoneme to voiceless. This happens in the phoneme space.

```
C:
    ...
    <rule devoice phonation> == [voiceless]
```

Without restrictions devoice would apply to any consonant. However, only obstruents (fricatives and plosives) may be devoiced in Dutch. Therefore, the rule devoice should test for the value of manner before taking effect. This results in the following extension[10]:

```
    <rule devoice phonation> == <test G_NEWOBJ:<manner>>
    <test [fricative]> == <test fricative G_NEWOBJ:<place>>
    <test fricative [uvular]> == [voiced]
```

[10]The CUT operator throws away the first element in the local path. It is used when there is extra information in the path that we do not need. CUT is defined as follows: CUT: <$any> == LocalPath:<>.

```
<test fricative> == [voiceless]
<test [plosive]> == [voiceless]
<test> == INPUT:<phonation CUT:<>>
```

This is the DATR code which makes final devoicing happen. For illustratory purposes, we include an output trace for the query `Hand:<singular phon syll 1 rhyme coda 2 phonation>` in the SS/BODYPART implementation. Here we query our sample fragment for the value of the feature `phonation` of the second element of the coda of the Dutch lexeme *Hand*.

```
== Hand:<singular phon syll 1 rhyme coda 2 phonation> ?
```

into Lexeme Module

```
** Call :  Hand:<singular phon syll 1 rhyme coda 2 phonation>?
** Call :  RULE:<singular phon syll 1 rhyme coda 2 phonation>?
** Call :  Hand:<rule singular phon syll 1 rhyme coda 2 phonation>?
** Call :  INPUT:<phon devoicelastsylllastcoda syll 1 rhyme coda 2 phonation>?
** Call :  Hand:<phon devoicelastsylllastcoda syll 1 rhyme coda 2 phonation>?
```

into Syllable Sequence Module

```
** Call :  RULE:<devoicelastsylllastcoda syll 1 rhyme coda 2 phonation>?
** Call :  INPUT:<focus syll 1 rhyme coda 2 phonation>?
** Exit :  INPUT:<focus syll 1 rhyme coda 2 phonation> = 1?
** Call :  RULE:<devoicelastsylllastcoda syllrule syll 1 rhyme coda 2
phonation>?
** Exit :  RULE:<devoicelastsylllastcoda syllrule syll 1 rhyme coda 2
phonation> = devoicelastcoda?
** Call :  INPUT:<syll 1 devoicelastcoda rhyme coda 2 phonation>?
```

into Syllable Module

```
** Call :  RULE:<devoicelastcoda rhyme coda 2 phonation>?
** Call :  INPUT:<rhyme coda subtype>?
** Exit :  INPUT:<rhyme coda subtype> = 2?
** Call :  RULE:<devoicelastcoda codarule>?
** Exit :  RULE:<devoicelastcoda codarule> = devoice?
** Call :  INPUT:<rhyme coda 2 devoice phonation>?
```

into Phoneme Module

```
** Call :  RULE:<devoice phonation>?
** Call :  RULE:<devoice manner>?
** Call :  INPUT:<manner>?
** Exit :  INPUT:<manner> = [plosive]?
** Exit :  RULE:<devoice phonation> = [voiceless]?
Hand:<singular phon syll 1 rhyme coda 2 phonation> = [voiceless].
```

This trace shows how the rule `devoice` in the phoneme space gets invoked via a series of rules in the higher modules. In the lexeme module, the rule `singular` invokes the rule `devoicelastsylllastcoda`, which in the syllable sequence module invokes the rule `devoicelastcoda` which in the syllable module invokes the rule `devoice`. In the phoneme module, the rule `devoice` tests for the value of `manner` of the last coda and as this is a `plosive`, changes the value of the feature `phonation` to `voiceless`.

Our second example considers how a series of rules can be invoked together, using Danish nouns. As in most languages, Danish nouns can occur in singular and plural, but in addition a definite article can be added to the end of the singular or plural form, e.g. *mund* ('mouth') - *munden* ('the mouth'); *munder* ('mouths') - *munderne* ('the mouths'). In the singular the definite article depends on the gender of the noun: *-et* is added to neuter nouns, *-en* to non-neuter nouns. In the plural, first the ending *-er* is added, followed by the definite article *-ne*. The operation of these two rules together in our framework is sketched in Figure 5.8[11] . In this example, the rule `definite` applies to the output of `plural` and the rule `plural` applies to the input word form structure.

The DATR code for the rule `plural` and the rule `definite` in Danish is as follows:

```
PLURAL_NOUN_ER_RULE:
    <> == PLURAL_NOUN_RULE
    <phon> == Suffix_O_o:<>
    <phon concat 1> == INPUT:<phon schwadeletion>.

DEFINITE_NOUN_RULE:
    <> == INPUT
```

[11] `/O_o/` stands for lowered `/O/`.

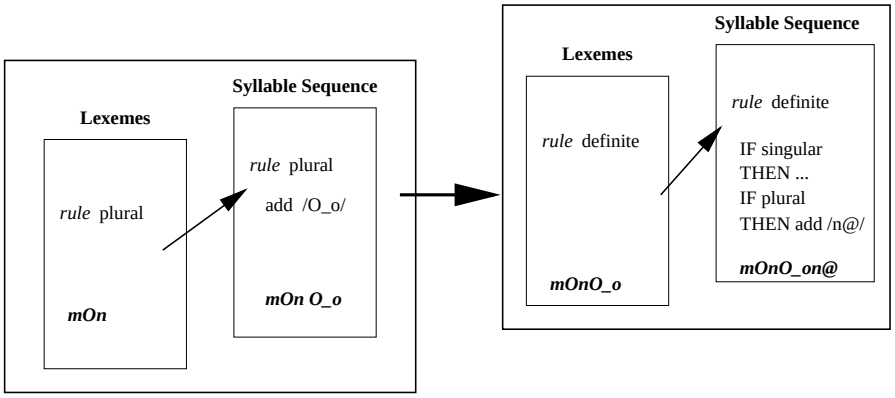


Figure 5.8: Plural and definite rules applied to the Danish lexeme *Mund* /mOn/ ('mouth')

```
<phon> == <test INPUT:<number>>
<test plur> == DK_Suffix_n@:<>
 ...
<phon concat 1> == INPUT:<phon>.
```

The DEFINITE_NOUN_RULE needs to test for number as different suffixes apply in singular and plural. The effect of both rules is that a suffix is added to the input phonological form. A suffix is defined as:

```
SUFFIX:
    <> == CONCAT
    <concat subtype> == 2
    <concat 1> == INPUT:<phon>.

CONCAT:
    <> == RULE
    <type> == concatenation
    <concat subtype> == 0.
```

SUFFIX inherits from CONCAT which is a simple left-right concatenation of two elements, here the input phonological form (in the case of the plural form, this is the base form; in the case of the plural definite form, this is the plural form) and a suffix.

We now consider the output trace for the query Mund:<plural definite phon

concat 2 syll 1 rhyme peak 1 height> in the SS/BODYPART implementation focussing on the working of the rule mechanism. Here we look for the value of the feature height of the peak of the last syllable of the form *mundern<u>e</u>*.

```
== Mund:<plural definite phon concat 2 syll 1 rhyme peak 1 height> ?
** Call :  Mund:<plural definite phon concat 2 syll 1 rhyme peak 1 height>?
** Call :  RULE:<plural definite phon concat 2 syll 1 rhyme peak 1 height>?
** Call :  RULE:<definite phon concat 2 syll 1 rhyme peak 1 height>?
** Call :  Mund:<rule definite phon concat 2 syll 1 rhyme peak 1 height>?
** Call :  INPUT:<number concat 2 syll 1 rhyme peak 1 height>?
** Call :  PREVRULE:<plural definite phon concat 2 syll 1 rhyme peak 1 height>?
** Call :  PREVRULE:<definite phon concat 2 syll 1 rhyme peak 1 height>?
** Exit :  PREVRULE:<definite phon concat 2 syll 1 rhyme peak 1 height> = ?
** Exit :  PREVRULE:<plural definite phon concat 2 syll 1 rhyme peak 1 height>
= plural?
** Call :  Mund:<plural number concat 2 syll 1 rhyme peak 1 height>?
** Call :  RULE:<plural number concat 2 syll 1 rhyme peak 1 height>?
** Call :  Mund:<rule plural number concat 2 syll 1 rhyme peak 1 height>?
** Exit :  Mund:<rule plural number concat 2 syll 1 rhyme peak 1 height>
= plur?
** Exit :  Mund:<rule definite phon concat 2 syll 1 rhyme peak 1 height>
= [midlow]?
Mund:<plural definite phon concat 2 syll 1 rhyme peak 1 height> = [midlow].
```

We see in this trace, how the rule mechanism gets the last rule of the sequence first, i.e. definite, and tries to apply it. When looking for its input, the rule definite finds that it needs the output of some previous rule as its input. PREVRULE returns the name of this rule, plural. The rule plural applies and its output forms the input of the rule definite. We can then determine the value of the height feature of the last peak of mundern**e**, which is [midlow].

Our final example considers a multilingual rule definition – the rule for nouns with a plural ending in *-s* in English, Dutch, and German. In our framework, this fact will be captured by a lexical rule which adds an *-s* suffix to the root of the noun. This *-s* suffix is realised as an /s/ in Dutch and German, and as an /s/ or /z/ in English due to voicing alternation. In all three languages a vowel is inserted before the *-s*, if the root ends in a sibilant. This vowel is realised as a /@/ in Dutch

and German, and as an /I/ in English. The definition of the plural_s rule in our framework is illustrated schematically in Figure 5.9 below for the Structure-Sharing model.

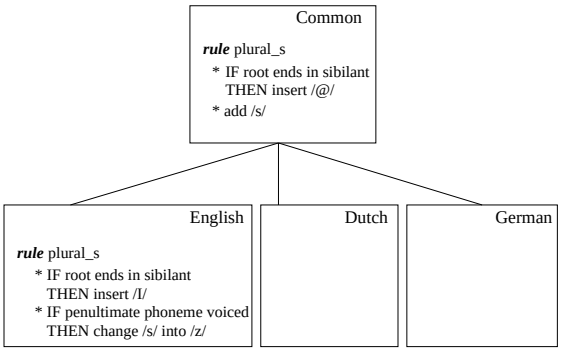


Figure 5.9: Definition of plural_s rule

Dutch and German inherit the plural_s rule as it is defined in the common part. The English plural_s rule inherits from the common part, but overrides the value of the suffix peak, using /I/ instead of /@/. It also adjusts the voicing of the final /s/ depending on the voicing of the preceding phoneme.

A possible implementation of the plural_s rule in DATR is as follows:

```
PLURAL_NOUN_S_RULE:
    <> == INPUT
    <number> == plur
    <phon> == Suffix_s:<>.

Suffix_s:
    <> == SUFFIX:<>
    <concat 2> == MONOSYLLABLE:<>
    <concat 2 syll 1> == "S_s:<assimilate NOUN_ENDING_S:<>>".

NOUN_ENDING_S:
    <> == <test coda EQ:<0 INPUT:<phon syll INPUT:<phon syll
                              subtype> rhyme coda subtype>>>
    <test coda false> == <test coda sibilant
           LASTSYLLLASTCODA:<manner> LASTSYLLLASTCODA:<place>>
    <test coda sibilant '[fricative]' '[apical]'> == @
```

```
    <test coda sibilant '[fricative]' '[palato]'> == @
    <test coda> == s.

S_s:
    <> == Syllable_xC
    <rhyme coda 1> == "C_s:<>"
    <rule assimilate @ rhyme peak subtype> == 1
    <rule assimilate @ rhyme peak 1> == "V_@:<>"
    <rule assimilate> == <CUT:<>>.
```

The node `PLURAL_NOUN_S_RULE` adds a suffix *-s* to the input. The exact realisation of the suffix *-s* depends on the output of `NOUN_ENDING_S`, which tests for the value of the last element of the root. First, it determines whether there is a coda or not by testing if the value for `subtype` equals 0 (line 1). If there is a coda, i.e. the `EQ` function fails, `NOUN_ENDING_S` tests whether this coda is a sibilant or not by determining its values for the features `manner` and `place` (line 2). If the last coda is a sibilant, i.e. `manner` is `fricative` and `place` is `apical` or `palato`, it outputs a @ (line 3 and 4), otherwise an `s` (line 5). The result is that when `NOUN_ENDING_S` outputs a @, a peak is added to the syllable `S_s`, realising it as `/@s/` instead of `/s/`.

English inherits its definition of the `plural_s` rule from the multilingual definition, except for the value of the peak inserted after a sibilant, which is an `/I/`, and for voicing alternation. The definition of the English `E_PLURAL_NOUN_S_RULE` is given below.

```
E_PLURAL_NOUN_S_RULE:
    <> == PLURAL_NOUN_S_RULE
    <phon> == E_Suffix_s:<>.

E_Suffix_s:
    <> == Suffix_s
    <concat 2 syll 1> == "E_S_s:<assimilate E_NOUN_ENDING_S:<>>".

E_NOUN_ENDING_S:
    <> == NOUN_ENDING_S
    <test coda sibilant> == <test coda phonation
```

```
                LASTSYLLLASTCODA:<phonation> CUT:<CUT:<>>>
    <test coda phonation> == z
    <test coda phonation '[voiceless]'> == s
    <test coda sibilant '[fricative]' '[apical]'> == I
    <test coda sibilant '[fricative]' '[palato]'> == I
    <test coda true> == z.

E_S_s:
    <> == S_s
    <rule assimilate I rhyme peak subtype> == 1
    <rule assimilate I rhyme peak 1> == "V_I:<>"
    <rule assimilate I rhyme coda 1> == "C_s:<>"
    <rule assimilate z rhyme coda 1> == "C_z:<>".
```

5.5 Summary

This chapter described the lexical description framework that is used in the sample lexicons of this thesis. The framework covers morphological and phonological aspects of word formation. Using a highly modular default-inheritance based approach, it supports the description of morphological and phonological lexical generalisations in a uniform representation system making no sharp distinction between morphology and phonology. The lexicon is organised into distinct self-contained modules corresponding to levels of lexical description (lexemes, syllable sequences, syllables, and phonemes). Each module makes extensive use of lexical rules to represent higher level relationships between word forms. Consequently the framework provides a flexible means for capturing linguistic generalisations within and across languages at all levels of lexical description that the framework distinguishes. The lexical description framework has been implemented in DATR.

Chapter 6

Implementation and Evaluation

6.1 Introduction

This chapter discusses the implementation of the sample lexicons and presents an evaluation of the results. The chapter is organised as follows. First, an overview is given of the conventions that have been used in the implementation in general. Then we discuss the implementation of the different models separately and finally we turn to their evaluation.

6.2 The implementation in general

The sample lexicons for the different architectures have been implemented and tested running Sussex/Brighton DATR-2.8 under Sussex Poplog Prolog. The body part test set has been implemented in the Structure-Sharing model, the Meta-Features model, and the Infinitesimal model. The medical test set has been implemented in the Structure-Sharing model and the Meta-Features model. Only a small fragment has been implemented for the MetaTheory model. The sample fragments were implemented in the following order. The Structure-Sharing model was implemented first adopting a non-parallel development strategy. This way all the data was made available and studied and this information could then be used for the implementation of the parameterised Meta-Features and Infinitesimal models. For each model, the body part test set was implemented first. The code of the body part fragments is included in Appendix **??**.

In the implementation of the sample lexicons, we use the lexical description framework discussed in the previous chapter. As we saw, this is a highly modular description framework which distinguishes four levels of linguistic description – lexeme module, syllable sequence module, syllable module, and phoneme module. In each of those, information can be shared in the multilingual lexicon and it is possible to treat each module as a separate network without worrying about alignment between the modules. In our multilingual lexicons, we adopted a fairly unified data-driven approach to information sharing across the different modules in that we generally share information which occurs in more than one language. Let us consider how information is shared in the different modules in the implementation of the different architectures.

Information sharing in the lexeme module

In the Structure-Sharing model, morphological properties are shared in the lexeme module if they occur in more than one language. For example, both Dutch and English have a plural noun ending in *-s* and this morphological property will therefore be defined in the shared part. However, the specific realisation of the noun ending in *-s* will be defined in the language-specific parts as there are differences between the two languages. In the parameterised models, all noun classes are defined in a single hierarchy.

Information sharing in the syllable sequence module

The syllable sequence module defines the number of syllables that make up a particular lexeme and inherits the actual syllables from the syllable module. We found in our test data that words which come from a single root tend to consist of the same number of syllables and if not there was often a two-way split between the four languages in the test set (but not consistently enough between the languages to introduce subhierarchies). Therefore, only information which occurs in the majority of the languages is shared in the syllable sequence module. For example, on the basis of the transcriptions in Table 6.1, the shared lexical entry for *Diameter* will be defined as a foursyllable in all models.

The same approach is adopted for the sharing of the syllable information in the lexical entries in the Structure-Sharing and Meta-Features models. Syllables are only defined in a multilingual lexical entry if an identical syllable (i.e. consisting of exactly the same phonemes in the same order) occurs in the same position in the same lexical entry in more than one language. For example, the multilingual lexical entry for *Diameter* (see Table 6.1) will have shared values for its third and

	English	Dutch	Danish	Icelandic
Diameter	d a I	d i:	d i	T v E r
	{	j a:	a	m A u l
	m I	m e:	m e:	
	t @ r	t @ r	d O_o	

Table 6.1: Phonemic transcriptions for the lexical entry *Diameter*

fourth syllables. In the Infinitesimal model and the MetaTheory model, on the other hand, language-specific syllables that occur in the same position in the same lexical entry are grouped together in multilingual syllable definitions. For example, the third syllable of the lexeme *Diameter* will have a multilingual syllable definition which has a cv skeleton and default values for the onset (/m/) and the peak (/e:/). A language parameter will be used in this syllable definition to realise the peak as an /I/ in English.

Information sharing in the syllable module

In the syllable module shared syllable inventories are defined. In the Structure-Sharing model, this shared syllable inventory consists of identical syllables which occur in more than one language regardless of which lexical entries they occur in and in which syllable position. For example, English, Dutch, and Danish all have a syllable /dEn/ but this syllable does not necessarily occur in the same lexical entries in the three languages. In our implementation, the syllable /dEn/ will be defined once in a shared syllable inventory and then it will be inherited by the different lexical entries using this syllable in the three languages.

In the Meta-Features model all syllables are defined in a shared inventory, regardless of which lexical entries they occur in, syllable position and language. In the Infinitesimal and MetaTheory models syllable position and occurrence have been taken into account and syllables occurring in the same syllable position in

	English	Dutch	Danish	Icelandic
Perforation	p 3:	p E r	p { R	r 9 y: v
	f @	f o:	f o	
	r e I	r a:	R A	
	S n	t s i:	S o: n	

Table 6.2: Phonemic transcriptions for the lexical entry *Perforation*

the same lexical entry in the different languages are grouped together into multilingual syllable definitions, as was illustrated for the third syllable of the lexeme *Diameter*.

In all models, the shared syllable inventory is augmented by capturing identical syllables using metaphonemes (Tiberius and Cahill, 2000a; Tiberius and Cahill, 2000b), 'abstract' phonemes which capture cross-linguistic phoneme correspondences. The concept of metaphonemes was briefly introduced in Chapter 4 in Section 4.3.

The implementation also captures several monosyllabic affix correspondences (cf. Wolff, 1984) between Dutch, English, and Danish. An example of such a correspondence is the English affix /Sn/, which corresponds to /tsi:/ in Dutch and /So:n/ in Danish, as in the lexical entry for *Perforation* (see Table 6.2). This correspondence is captured by defining a multilingual syllable `S_Sn/tsii/Soon` which is realised as the syllable /Sn/ in English, /tsi:/ in Dutch, and /So:n/ in Danish.

Information sharing in the phoneme module

The general approach to information sharing in the phoneme module has been to define all the phonemes which occur in any of the languages of the test sets into one multilingual phoneme inventory. This seemed justified as most phonemes are used by more than one language of the test sets. The multilingual phoneme inventories have been extended with metaphonemes.

Limitations

Problematic for our syllable-based framework are those cases where there is not a one-to-one correspondence between the syllables in the different languages. This is for example the case for the lexeme *Elbow* which is a disyllabic word in English /El-b@U/ and a trisyllabic word in Dutch /E-l@-bo:x/ as illustrated in Table 6.3. Here there is an overlap between the first syllable in English and the first and second syllable in Dutch. The /l/ belongs to the first syllable in English, but to the second syllable in Dutch. This correspondence cannot be captured in our framework.

Dutch	**English**
/E/	/El/
/l@/	
/bo:x/	/b@U/

Table 6.3: Phonemic transcription for the lexical entry *Elbow*

In the implemented sample fragments only metaphonemes for vowels have been included. As we noted in Chapter 4 multilingual phoneme correspondences for consonants have not been studied yet and are therefore not included. Consequently, syllables that could potentially be shared are not shared in the implementation because of different consonant phonemes in the different languages. For example, if we take the lexeme *Contact* in Dutch, English, and Danish (see Table 6.4), then the current implementation will capture the similarities between the syllables /kQn/, /kOn/, and /kO_on/ containing the metaphoneme |QOO_o| and the syllables /t{kt/ and /tAkt/ containing the metaphoneme |{A|, but it will not capture the similarities between /t{kt/ and /tAgd/ where different but very similar consonant phonemes are used. Capturing metaphonemes for consonants can only further

	English	**Dutch**	**Danish**
Contact	k	k	k
	Q	O	O_o
	n	n	n
	t	t	t
	{	A	A
	k	k	g
	t	t	d

Table 6.4: Phonemic transcriptions for the lexical entry *Contact*

increase the amount of sharing in the sample fragments.

6.3 Implementation of the multilingual architectures

6.3.1 The Structure-Sharing Model

In Chapter 3 we saw that in the Structure-Sharing model, a multilingual lexicon is constructed by taking the monolingual hierarchical lexicons for each of the languages in the lexicon and creating parallel hierarchies which contain the information that the monolingual hierarchies have in common.

Both sample fragments were implemented in the Structure-Sharing model, i.e. the small test set of 19 body part terms, henceforth SS/BODYPART, and the larger test set of 100 medical terms, henceforth SS/MED.

The sample fragments were constructed as follows. A non-parallel development strategy was adopted to construct SS/BODYPART. First monolingual hierarchical lexicons were created for each of the languages in the lexicon – Dutch, English, Danish, and Icelandic – using the lexical description framework described in Chapter 5 and then the commonalities between these hierarchies were captured in shared hierarchies. The implementation of SS/MED was subsequently based on the available structure of SS/BODYPART. The main differences are that in the larger fragment, adjectives are introduced, more noun classes are distinguished, larger syllable sequences occur (up to 7 syllables), and more lexical rules are used.

An extract of the hierarchical structure of the SS/BODYPART lexicon is given in Figure 6.1. Figure 6.1 only contains part of the inheritance hierarchy (similar inheritance hierarchies exist for the syllable sequence, syllable, and phoneme modules), but it is already clear from this picture that there is a lot of redundancy in this network. Each language has its own hierarchy and the inheritance pattern within a language is repeated over and over again. For example, adding a new dialect, which is related to one of the languages already available in the lexicon, requires the establishment of a new parallel hierarchy with appropriate links to the parent language. This is illustrated in Figure 6.2 for a dialect of English. Thus, the Structure-Sharing model does not allow us to exploit non-monotonicity to the full, i.e. it is not very good for capturing minor variations.

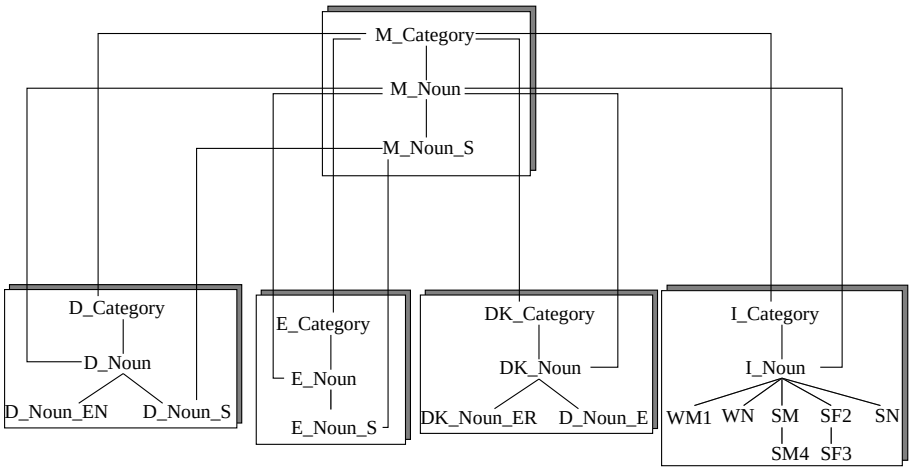


Figure 6.1: Lexeme hierarchy in the SS/BODYPART lexicon

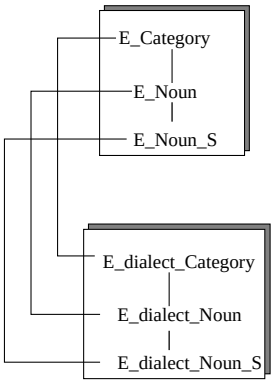


Figure 6.2: Adding a dialect to a Structure-Sharing model

Our fragment only covers a small set of lexical entries and one can imagine that when the lexicon becomes bigger and more languages are involved, the hierarchical structure and the interactions between the different hierarchies would become even less transparent. Because of the wealth of inheritance links, it is not always clear how the inheritance relations go. For example, in our fragment, there are two routes from `D_NOUN_S` to `M_NOUN`. You can go from `D_NOUN_S` via `D_NOUN` to `M_NOUN` or via `M_NOUN_S`. In the sample fragments, plural information is inherited directly from `M_NOUN_S`, whereas all other information is inherited via the `D_NOUN`, `M_NOUN` link.

There is no language parameter in the Structure-Sharing model. Each lexeme has a lexical entry per language, and the multilingual lexicons are queried per lexeme entry per language. For example, `Mouth` is the lexical entry for *mouth* in English, and `Mond` is the lexical entry for *mond* ('mouth') in Dutch. There is not a multilingual lexical entry node which you can query for Dutch, English, Danish, or Icelandic. That means, that the Structure-Sharing model does not force us into cross-linguistic identity claims at the lexeme level. In our fragments, this would not have been a problem, as we started from translation equivalents. However, we saw earlier that words with common morphology and phonology do not necessarily share their semantics, e.g. English *keen*, Dutch *koen*, and German *kühn*. Which cross-linguistic lexeme entry would we want to query in such a case? To keep track of which language we are dealing with, node names have been prefixed with a language abbreviation in the language-specific (lexeme, syllable sequence, and syllable) hierarchies.

6.3.2 General comments on the implementation of the parameterised models

Before turning to the implementation of the parameterised models separately, we first discuss a few general points relating to their implementation. We saw in Chapter 3 that in the parameterised models language features are inserted in the main feature theory. Our lexical description framework divides the feature space into two parts: a rule part and an object part. The parameterised models add a language part to this. If language features occur at the end of the rule-object part, DATR's path extension can be used to express inheritance relations within the language tree. If language features occur anywhere else in the rule-object part,

inheritance relations within the language tree cannot automatically be captured in DATR using path extension. To make inheritance work in the language tree as well as in the rule-object feature tree, an extra layer has been added to the implementation analogous to the implementation of the lexical rule mechanism in DATR. The rule mechanism was discussed in Section 5.3. The language mechanism is encoded as follows:

```
LANG:
        <$language> == <>
        <> == IN:<>.

IN:
        <> == "< "PREVLANG" >".

PREVLANG:
        <$language1 $language2> == $language1 <$language2>
        <$language> == .
```

At each level in the analysis, `LANG` strips off the language prefixes and passes the rule-object path to `IN`. `IN` takes the rule-object path as its input and prepends it with the language string based on the global context. If the resulting language-rule-object path cannot be analysed, the rightmost language is stripped off (`PREVLANG` returns all the languages except for the last one) from the language part until a match is found. For example, if we have a language path <`indoeuropean germanic west dutch`>, then we first look for a match of <`indoeuropean germanic west dutch`> followed by the rule-object path. If no match is found, `dutch` is stripped off and we try to find a match for <`indoeuropean germanic west`>-rule-object path and so on. When a match is found, Dutch will inherits its definition from there. Our sample fragments use a rather flat language typology without subtrees. The language tree is illustrated in Figure 6.3.

In the parameterised models, all information is integrated into one single hierarchy. The question that arises is how should this hierarchy be structured? What can be defined as default? Consider, for example, the lexeme module in our sample fragments. How do we define the multilingual noun class hierarchy, i.e. how do we determine which plural noun endings can be grouped together cross-linguistically? At least the hierarchies shown in the figures 6.4 and 6.5 can be defined for the major noun classes in Dutch, English, and Danish.

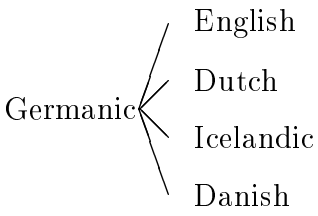


Figure 6.3: Language tree used in the implementation

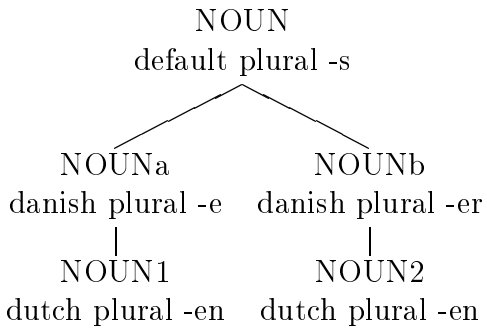


Figure 6.4: Noun Hierarchy 1

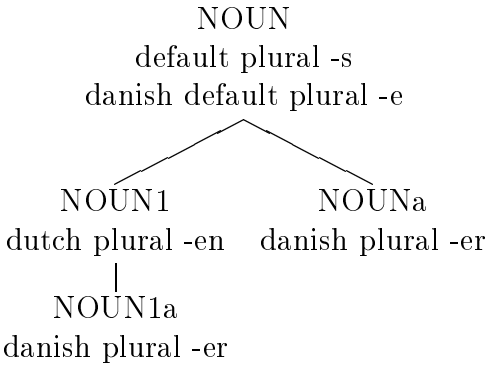


Figure 6.5: Noun Hierarchy 2

The situation gets even more complicated when Icelandic is incorporated for which eight noun classes are distinguished in the sample fragments. The problem we are confronted with here is fundamentally a linguistic one. More cross-linguistic research seems necessary to provide insight in which hierarchy is linguistically justified. If, however, we are not concerned about linguistic viability, machine learning techniques (Lock, 2000) could be used to automatically construct a single hierarchy.

6.3.3 The Meta-Features Model

In the Meta-Features model, language parameters are expressed as meta-features. That is, each feature-value path in the hierarchy has a full language parameter setting associated with it. This parameter setting is embedded in the feature space by prefixing it to the ordinary lexical entry specifications. In our implementation of the Meta-Features model this means that language parameters are inserted before the rule part, making lexical rules language-specific. Since lexical rules can be invoked within each module, language parameters can be introduced before lexical rules within each module in our lexical description framework. This situation is illustrated in Figure 6.6. Rules without a language parameter are considered to be

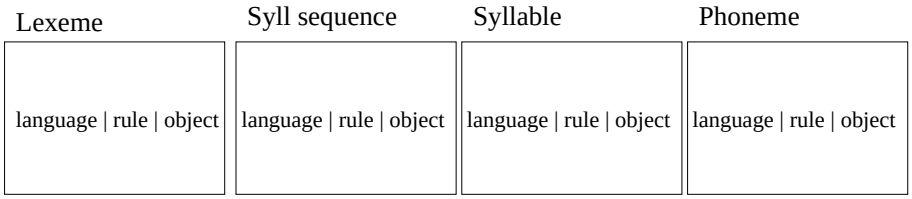


Figure 6.6: Feature space in the Meta-Features Model

language-neutral.

Both test sets were implemented in the Meta-Features model. They are referred to subsequently as META/BODYPARTS for the body part fragment and META/MED for the medical fragment. As SS/MED built on the structure of SS/BODYPART, META/MED built on the available structure of META/BODYPART. The main differences are that META/MED contains adjectives, larger syllable sequences and more lexical rules. However, rather than trying to integrate the Icelandic noun classes into the shared noun hierarchy as was done for the body part fragment, the noun classes for Icelandic are primarily specified in the individual

lexical entries in the implementation of the medical test set. Manually restructuring the noun hierarchy proved too difficult on the basis of 25 Icelandic lexical entries. Both fragments were constructed using a parallel development strategy.

We illustrate now how the Meta-Features model works. When a lexical rule is encountered, the lexical rule mechanism checks first whether language parameters are present and tries to find a match for a path consisting of the whole language sequence followed by the lexical rule. If there is no match for this path, the lexical rule mechanism strips off the rightmost language and tries to apply the resulting language-rule path again and does this until a match is found. Assume that we are looking for the English definition of the rule singular. In our sample fragment, the lexical rule mechanism will first check whether the rule is defined for <`germanic english`> – on the basis of our language typology – and if not, it will strip off `english` and check whether the rule `singular` is defined for <`germanic`>. If still no match is found, `germanic` is stripped off and the lexical rule mechanism looks for a definition of the rule `singular` without a language parameter.

The situation gets more complex when a larger typology is used and when a sequence of rules occur. For instance, assume that we have a language typology in which `danish` inherits from `germanic` which inherits from `indoeuropean` and a sequence of rules `singular` and `definite`. The lexical rule mechanism gets the rightmost rule first, puts the language parameters back in and tries to find a match, stripping off `danish`, `germanic`, and `indoeuropean` if necessary. The rule `definite` gets its input from rule `singular` and the whole analysis is repeated for rule `singular`. The rule mechanism puts the language parameters back in and progressively strips them off until a match is found and output is returned. The output of the rule `singular` is then used by the rule `definite` to realise its output.

Lexical rules can be invoked in each module of our lexical description framework and thus rules can be made language-specific in each module. The rules we saw so far (`singular` and `definite`), were defined in the lexeme module. As an example, let us illustrate how the devoicing rule in the phoneme module can be made language-specific. The devoicing rule was defined in Section 5.4 for Dutch. In Dutch, final devoicing only applies to obstruents, but we could imagine that it has a wider scope in other languages. In that case, the definition of the rule `devoice`, which is repeated here, would change as follows:

```
C:
    ...
    <rule devoice phonation> == [voiceless]
    <germanic dutch rule devoice phonation> == <test G_NEWOBJ:<manner>>
    <test [fricative]> == <test fricative G_NEWOBJ:<place>>
    <test fricative [uvular]> == [voiced]
    <test fricative> == [voiceless]
    <test [plosive]> == [voiceless]
    <test> == INPUT:<phonation CUT:<>>.
```

The statement (second equation) which checks whether an obstruent is involved is made language-specific and now only applies to Dutch.

The fact that language parameters can be inserted before lexical rules in each module, makes our implementation of the Meta-Features model more powerful than the abstract model defined in Chapter 3, which would be equal to allowing a language parameter in the lexeme space but not in any of the other modules.

6.3.4 The Infinitesimal model: A restricted version

In the Infinitesimal model, a language feature can in principle occur anywhere in the feature-value path, at the beginning, at the end, and anywhere in between. This makes the Infinitesimal model potentially a very powerful model as cross-linguistic generalisations can be captured at all levels of granularity.

In this thesis, we implemented a restricted version of the Infinitesimal model in which a language feature can be inserted before the lexical rule part and before the object part in each module of our lexical description framework. This situation is illustrated in Figure 6.7.

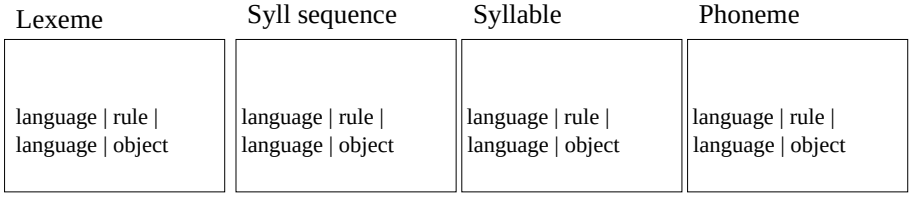


Figure 6.7: Feature space in the Infinitesimal Model

Lexical rules are made language-specific in this model, in the same way as in the Meta-Features model and will therefore not be discussed again. On top of this, the object parts can be made language-specific by allowing language-features to occur at the beginning of the object part of each module. Let us illustrate how this works for the implementation of the phonological forms in our framework. Here we have a possible definition of the lexeme *Arm*:

```
%% LEXEME MODULE

Arm:
        <> == NOUN_SMa
        <phon> == "X_ARM:<LANGPART:<QueryPath>>".

%% SYLLABLE SEQUENCE MODULE

X_ARM:
    <> == MONOSYLLABLE
    <germanic icelandic> == DISYLLABLE
    <germanic syll 1> == "S_AAm/Arm:<LANGPART:<QueryPath>>"
    ... .

%% SYLLABLE MODULE

S_AAm/Arm:
    <> == Syllable_VC:<>
    <germanic dutch rhyme coda subtype> == 2
    <germanic rhyme peak 1> == "V_AA:<>"
    <germanic dutch rhyme peak 1> == "V_A:<>"
    <germanic rhyme coda 1> == "C_m:<>"
    <germanic dutch rhyme coda 1> == "C_r:<>"
    <germanic dutch rhyme coda 2> == "C_m:<>"
    ... .
```

Language parameters are passed on in the feature-value paths between the different modules. Here, the lexeme *Arm* inherits its phonological form from `X_ARM` and language features are passed on to the syllable sequence module in the feature-value path of `X_ARM`. `X_ARM` then inherits its syllable definition from `S_AAm/Arm` and

by inserting language parameters in the feature-value path of `S_AAm/Arm`, they are passed down to the syllable module. This way cross-linguistic generalisations at the object level can be captured in each module. In this fragment, we see how default information in the syllable sequence and syllable modules is overridden. The syllable sequence information is overridden for Icelandic where the lexical entry for *Arm* is a disyllable, and the syllable information needs to be overridden for Dutch and Icelandic to obtain the Dutch syllable /`Arm`/ and the Icelandic syllable /`Ar`/.

By allowing a language parameter to occur before the object part in each of the modules describing the phonological form of a lexeme, semantically related information can be grouped which is not necessarily morphologically and/or phonologically related. In the case of the lexeme *Arm*, there was enough morphological and phonological similarity to warrant the approach. Consider, however, a shared syllable definition for the lexeme *Curve* in English /`k3:v`/ and Icelandic /`hnI:t`/.

```
S_k33v/hnIIt:
        <germanic english> == Syllable_CVC
        <germanic icelandic> == Syllable_CCVC
        <germanic english onset 1> == "C_k:<>"
        <germanic icelandic onset 1> == "C_h:<>"
        <germanic icelandic onset 2> == "C_n:<>"
        <germanic english rhyme peak 1> == "V_33:<>"
        <germanic icelandic rhyme peak 1> == "V_II:<>"
        <germanic english rhyme coda 1> == "C_v:<>"
        <germanic icelandic rhyme coda 1> == "C_t:<>".
```

Do we really want to group this information together? There is no shared element here. We feel that more cross-linguistic knowledge is required to make linguistically grounded decisions on where cross-linguistic identity claims should be allowed. Therefore, only the body part test set was implemented in the variant of the Infinitesimal, where a language feature can occur before the rule part and before the object part in each module. A parallel development strategy was used.

6.3.5 The MetaTheory Model

The MetaTheory model is conceptually a totally different model from the other multilingual architectures explored in this thesis. The multilingual lexicon does not consist of a set of lexicons, but of a set of default inheritance descriptions of monolingual lexicons plus a universal metatheory describing the formalism that is used (e.g. DATR). Monolingual lexicons can be compiled out of the multilingual lexicon by querying the lexicon for the object theory of a particular language.

The MetaTheory model is quite unconstrained. There is nothing in the model which tells you what the metatheory for a set of languages should look like. Inheritance relations can be captured at both levels, the metatheory level and the object theory level. Should inheritance relations be captured at both levels, and if so should they follow the same pattern? If a set of monolingual hierarchical lexicons is available for the languages for which a metatheory is built, then the inheritance relations in the metatheory can be guided by these monolingual hierarchical structures. But how do you decide on a hierarchy if such monolingual hierarchical structures are not available?

Because of these theoretical issues, only a small fragment was implemented. We created a metalexicon for a few Dutch and English words to test the potential of the MetaTheory model. By querying this metalexicon for Dutch, a monolingual Dutch lexicon is generated, by querying it for English, a monolingual English lexicon is generated. DATR's definition by default mechanism means that the language features which occur in the Query path will automatically be passed on through the different levels in the metalexicon. This makes the MetaTheory model as powerful as the Infinitesimal model allowing language parameters to occur with the rule part and with the object part in each module of our lexical description framework. Let us see how this works in practice.

Figure 6.8 gives an extract of the hierarchical structure of the metatheory describing the Dutch and English lexicons[1]. Each `METALEXICON` consists of a `METATHEORY` and `METALEXENTRIES`. `METATHEORY` contains higher level linguistic information. It contains variable definitions, definition of the infrastructure and a metalevel implementation of the different modules that our lexical description framework distinguishes. A lot of this information is language-independent. `METALEXENTRIES` defines the actual lexeme entries, their phonological forms, and syllables. At each

[1] We did not attempt a metalevel implementation of the DATR syntax in our sample fragment.

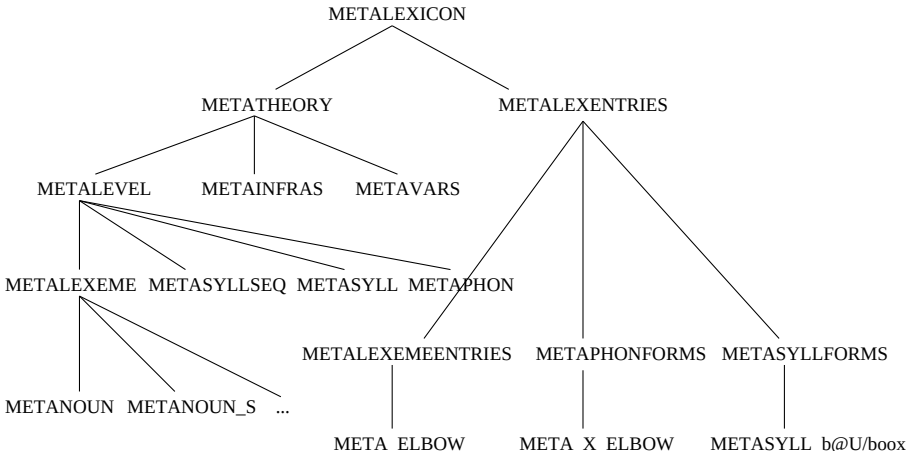


Figure 6.8: Extract of the inheritance hierarchy of the MetaTheory fragment

level in this hierarchy, language can be invoked. For instance, the node `METATHEORY` can generate an object theory specific to Dutch and an object theory specific to English, the node `METAVARS` can generate variable definitions specific to Dutch and variable definitions specific to English, and so on.

Differences in the object theories for the different languages are captured by invoking a language parameter in the appropriate metadefinition. For example, the noun class `NOUN_EN` only occurs in Dutch. The metanode describing this noun class should thus only generate output for Dutch and not for English. This is achieved by invoking a language parameter in the metadefinition of `NOUN_EN`. Recalling the definition of `NOUN_EN` in Section 5.4, the node `METANOUN_EN` could be defined as follows in its simplest form:

```
METANOUN_EN:
    <> ==
    <dutch> == 'NOUN_EN: <> == NOUN.'
               'NOUN_EN: <rule plural> == PLURAL_NOUN_EN_RULE:<>.' .
```

In the case of English, no output is generated. In the case of Dutch, the original `NOUN_EN` definition is returned.

In our fragment, no inheritance relations exist between the language parameters. We only used `dutch` and `english`. However, the MetaTheory model can also deal with a situation where the language parameters are linked together in a language typology. As the tree structure of the object metatheory is the same for

all languages in the lexicon, DATR's path extension will automatically capture inheritance relations in the language tree if present.

6.4 Evaluation

In this section we evaluate the sample fragments that were implemented for the Structure-Sharing, The Meta-Features, and Infinitesimal models. The MetaTheory fragment is not included in the evaluation because of the theoretical issues involved in its implementation. The evaluation of the sample lexicons is based on the criteria defined in Chapter 4. For ease of reference, they are summarised below.

For each of the sample fragments, we determine its validity and quality. We assume that a sample fragment is valid if it is consistent and complete with respect to the input data.

In order to determine which of the sample fragments is best, we evaluate the quality of the DATR theories by using the following criteria:

- Size of the DATR theory
  The size of a DATR theory is measured by the average number of sentences per object. Two counts were done, one where an object is any node, and one where only lexical nodes are objects. The first calculation gives us a general idea of the size of a DATR fragment. The second calculation indicates how much information one needs to define a certain number of lexical entries (in our case, 19 in the body part test set and 100 in the medical test set).

- Inferential complexity
  The inferential complexity is measured by the number of steps that are required to look up a particular value in the lexicon.

- Extendability
  The extendability of a model is measured by determining how easy it is to change information in the lexicon and to add new words and languages to the lexicon.

- Flexibility with respect to development strategy
  Can the multilingual lexicon be implemented using a parallel or a non-parallel strategy or both?

Model	Number of statements per node
SS/BODYPART	1536/527 = 2.91
META/BODYPART	1430/427 = 3.35
INF/BODYPART	1183/301 = 3.9
SS/MED	5475/1594 = 3.43
META/MED	4483/1213 = 3.7
INF/MED	–

Table 6.5: Overview of the number of statements per node

Model	Number of statements per lexical node
SS/BODYPART	1548/19 = 81.47
META/BODYPART	1430/19 = 75.26
INF/BODYPART	1183/19 = 62.26
SS/MED	5475/100 = 54.75
META/MED	4483/100 = 44.83
INF/MED	–

Table 6.6: Overview of the number of statements per lexical node

First, we determined the validity of each of the sample fragments. All implemented sample fragments cover the data in the test sets and are therefore valid models. We then determined the quality of the sample implementations using the following measures:

- **Size of the DATR theory**

  To measure the size of a DATR theory, we counted the number of statements per node in each sample fragment as well as the number of statements per lexical node. The results are shown in Table 6.5 and 6.6.

  As noted earlier, there is a lot of repetition of information in the Structure-Sharing fragments. Each language has its own hierarchy, and thus each lexical entry has a corresponding node plus statements in each language. Therefore we expect more nodes, and less information per node in the Structure-Sharing model than in the parameterised models. In the parameterised models, all information is specified once in a single hierarchy. This way redundancy is avoided. Consequently, we expect the number of statements per lexical node to be lower in the parameterised models. The data in Table 6.5 and 6.6 confirm this.

Model	Inference Steps phonation of peak *Arm*
SS-dutch	38
SS-english	12
SS-danish	12
SS-icelandic	13
META-dutch	110
META-english	37
META-danish	26
META-icelandic	26
INF-dutch	176
INF-english	86
INF-danish	75
INF-icelandic	64

Table 6.7: Overview of inferential complexity

- **Inferential Complexity**

  To measure the inferential complexity, we counted the number of inference steps that were necessary to look up a set of values in the different sample fragments. Table 6.7 gives the number of inference steps that were necessary to look up the value of the phonation feature of the peak of the singular form of the lexeme *Arm* in all four languages[2].

  The parameterised models – Meta-Features and Infinitesimal– take far more inference steps than the non-parameterised model to look up a particular value. The reason for this is that in the parameterised Meta-Features and Infinitesimal models a language tree is inserted before the rule or the rule and object part in each module. Inheritance relations take place in the language tree and are processed at each step in the analysis. For each rule part and for each object part, the language tree needs to be evaluated. Consequently, the parameterised models perform worse on inferential complexity than the non-parameterised Structure-Sharing model. The number of inference steps is particularly high for Dutch in the Meta-Features and the Infinitesimal model because of the final devoicing rule which applies to Dutch singular forms.

[2]The query is something like `Arm:<singular phon syll 1 rhyme peak 1 phonation>`. The rules for definiteness in Danish and case in Icelandic have not been taken into account in this calculation.

- **Extendability**

  The extendability of a parameterised model depends largely on the structure of the multilingual hierarchy and the language typology that is used. Once a language typology is chosen and a multilingual hierarchy has been constructed, it is difficult to incorporate changes as everything depends on this hierarchy and language typology. For example, in the implementation, we used a flat language typology. If at some point, we would have wanted to group Danish and Icelandic together and Dutch and English, then we would have had to change the language typology to the tree structure represented in Figure 6.9. Additionally, we would have had to check all information which

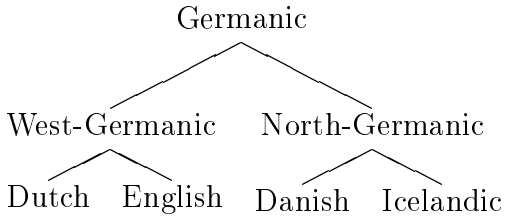


Figure 6.9: Revised language typology for our test sets

is specified in the lexicon to see whether it is true for all languages, for a particular subset of languages, or for a particular language. Similarly, changes in the multilingual object hierarchy affect the whole lexicon. If the hierarchy has been constructed 'correctly' for the languages in question, adding new words to any of the languages in the lexicon should not be a problem. For example, adding a lexeme entry for *Thigh* to our body part fragment, requires determining the noun class of the shared entry and defining its phonological form for all four languages similar to the definition of the phonological form of the lexeme *Arm* which was illustrated above. Adding new languages, however, could require complete restructuring of the hierarchy depending on whether the new languages are more or less distant to the languages already available in the lexicon.

The extendability of a non-parameterised model, on the other hand, does not depend on a language typology or the structure of the multilingual hierarchy. When a new word is added, it is compared to the information which is already in the lexicon, and if any information is shared by the majority of the languages, it is specified in the multilingual part. The integration of new languages requires setting up a language-specific inheritance network with

links to the shared network. Thus, the whole structure is less integrated than in a parameterised model. In the implementation, we experimented with the extendability of the SS/MED and META/MED fragments by including some Icelandic data in a later phase. In both fragments this was possible. In the Structure-Sharing model, a lot of extra structure had to be added to each of the modules distinguished in our lexical description framework. Apart from that, no major revisions were required. The shared noun hierarchy stays the same and the definition of the Icelandic phonological forms is mainly completely separate from the definition of the phonological forms in the other languages. The main difficulty encountered when adding the Icelandic data to the META/MED fragments turned out to be the reconstruction of the noun class hierarchy. This proved too difficult on the basis of such a small set of data (25 Icelandic nouns). Therefore, the noun class hierarchy was mainly left as it was and the Icelandic noun classes were primarily specified in the individual lexical entries.

Cahill (1998) discusses a method for the automatic acquisition of data for a Structure-Sharing lexicon, in particular the PolyLex lexicon. She describes an algorithm which automatically generated lexical entries for the PolyLex lexicon after the intial manual setting up of the hierarchy. Her algorithm is fairly simple and works as follows. It finds the maximum number of syllables. Then for each syllable up to the maximum it compares the values of the onset, peak, and coda in the three languages of the PolyLex lexicon. Where the maximum number of syllables is greater than the maximum number of syllables in one language, the remaining values are simply NULL. The next step is then to compare the array values and to abstract the generalities. The policy within PolyLex is to assume that where two or more languages share a piece of information this is the default. The algorithm was used to generate about 1300 noun entries and 1000 entries of other (non-verbal) categories in Dutch, English, and German. No such algorithm currently exists for the parameterised models.

- **Flexibility with respect to development strategy**
  We saw in Chapter 4 that roughly two development strategies can be used, parallel and non-parallel. Which development strategy is to be preferred depends on the amount of integration in the multilingual lexicon. Less integration allows for both a parallel and non-parallel development strategy,

whereas more integration favours a parallel development strategy. If all information is to be integrated into one single shared hierarchy, it is important to have all information available at the same time such that cross-linguistic generalisations can be captured immediately upon construction. Adopting a non-parallel development strategy would result in a lot of extra work. First all monolingual lexicons would be created separately and then this information would have to be integrated into one single hierarchy disposing of the structure of the monolingual lexicons. The parameterised models are generally more integrated and therefore prefer a parallel development strategy. The Structure-Sharing model is less integrated by nature as each language has its own hierarchy and the inheritance pattern is repeated for every language. Thus using a non-parallel development strategy is fine as the structure that is built for the monolingual lexicons is going to be incorporated in the multilingual Structure-Sharing lexicon.

Extension of the lexicon always involves non-parallel development. The fact that the parameterised models tend to be less easily extendable than the non-parameterised shows that they really do not like non-parallel development. The non-parameterised Structure-Sharing model does not mind, but the result might not always be optimal sharing.

6.5 Conclusion

In this chapter, we discussed the implementation of the sample fragments in the different models. First, an overview was given of the conventions that have been used in the implementation of the sample fragments in general. Then the implementation of the sample fragments was discussed separately for each model, before turning to their evaluation.

On the basis of the above discussion, we come to the conclusion that there is no single answer to the question what is the best way to structure a multilingual inheritance-based lexicon. All sample fragments cover the same input data, but in different ways. Each model has its advantages and disadvantages.

The appeal of the Structure-Sharing model is that it provides a rather straightforward way of constructing a multilingual resource from a set of monolingual lexicons using a uniform representation format. A multilingual structure sharing

lexicon is constructed by comparing the monolingual hierarchical lexicons for each of the languages and creating a parallel hierarchy containing what the monolingual hierarchies have in common. Apart from being rather straightforward, this procedure can also fairly easily be automated (as described in Cahill 1998) allowing the automatic construction of lexical resources for NLP applications.

The Structure-Sharing model is also a fairly robust model. Each language has its own hierarchy and language-specific changes can be easily incorporated without affecting the rest of the hierarchy. In the parameterised models, on the other hand, even minor changes can affect the whole hierarchy.

The Structure-Sharing model also performs better with respect to the inferential complexity. Inferential complexity is on average 4.14 times higher for the parameterised models than for the non-parameterised Structure-Sharing model.

The downside of the Structure-Sharing model is that there is a lot of redundancy. We saw in the evaluation that the Structure-Sharing model has 6.21 statements per lexical node more than the Meta-Features model and 19.21 more than the Infinitesimal model[3]. The reason for this is that each language has its own separate hierarchy (or set of hierarchies) and inheritance patterns are repeated over and over again. We saw that even if you would like to add a new dialect (related to one of the languages already available in the lexicon), a complete parallel hierarchy with appropriate links to the parent hierarchy needs to be established. As a consequence, the inheritance network in the Structure-Sharing model might be quite messy and therefore more difficult to maintain and extend.

The parameterised models avoid the kind of redundancy of the Structure-Sharing model. Parameterised multilingual lexicons consist of one single hierarchy in which a language parameter is used to conditionalise certain parts of the hierarchy for certain languages. In our approach, this language parameter is integrated in DATR's main feature theory which allows us to introduce language variation at different levels in the feature tree – before lexical rules in the Meta-Features model and before lexical rules and object parts in the Infinitesimal model and the MetaTheory model.

[3]The repetition of inheritance patterns is even more pronounced with the modular lexical description framework used in this thesis. This is because each language has its own lexeme hierarchy, syllable sequence hierarchy, syllable hierarchy, and phoneme hierarchy and for each lexical entry the appropriate values need to be defined for all of those for each language in the Structure-Sharing model.

Although the Meta-Features model and the Infinitesimal model seem to be able to describe the same data, the Infinitesimal model seems to be preferable as it allows us to capture generalisations that the Meta-Features model could not capture such as cross-linguistic generalisations at the object level in the different modules. We have shown, however, that it is not always self-evident from a linguistic perspective at which levels cross-linguistic generalisations are desirable. Recall the shared syllable definition for the lexeme *Curve* in English and Icelandic where no information was shared between the two languages. More cross-linguistic research could help to define which kinds of cross-linguistic generalisations are linguistically justified. It could also provide insight in how to construct a particular metatheory for a set of languages in the MetaTheory model.

Chapter 7

Concluding remarks

7.1 Contribution of the thesis

This thesis provides an in-depth analysis of different architectures that can be used to construct a multilingual inheritance-based lexicon in which information can be shared at different levels of linguistic description. In this thesis, we have focussed on the sharing of morphology, phonology, and morphophonology.

The architectures that we have explored in this thesis were defined in Chapter 3. Most of these architectures were inspired by proposals that were put forward by Evans (1996). Evans discussed three possible models – the non-parameterised Structure-Sharing model, the Micro-Features model, the Meta-Features model – and he hinted at a fourth model, the Infinitesimal model. The Infinitesimal model has matured into a full competitor in this thesis. We have investigated these four models and provided sample implementations in DATR where possible. In addition, a fifth model was studied, which is called the MetaTheory model.

On the basis of the analysis and implementation of the different models, we have gained insight into the research issues raised at the beginning of this thesis, viz.

- **The regulation of the inter- and intralanguage inheritance relations**
  That is, how do the inheritance relations go within and across languages in a multilingual inheritance network?
- **Multilingual information sharing**
  Does the choice of architecture affect the kind of information that can be shared and how is the information shared?

- **Development strategies**

  How does one go about constructing a multilingual inheritance lexicon? Should the monolingual and multilingual hierarchical lexicons be developed in parallel and linked immediately upon construction or should a non-parallel development strategy be adopted, where the monolingual lexicons are first fully developed separately and only linked together at the end?

Below we will summarise the answers to these research issues for the models investigated.

7.1.1 Regulation of the inter- and intralanguage inheritance

In a multilingual inheritance-based lexicon, there are inheritance relations which capture generalisations within a language and there are those that capture generalisations between languages. The question is how are these two kinds of inheritance relations expressed in the different models and how do they interact?

The sample implementations show that the parameterised and non-parameterised models behave differently with respect to the expression of inter- and intralanguage inheritance relations. In the non-parameterised Structure-Sharing model, there is a clear split between the inter- and intralanguage inheritance relations, whereas there is not such a clear split in the parameterised models. This is illustrated in the figures 7.1 and 7.2 by comparing the noun class hierarchy in the non-parameterised SS/BODYPART implementation with the corresponding hierarchy in the parameterised META/BODYPART implementation.

In the Structure-Sharing model, each monolingual lexicon keeps its own inheritance hierarchy modelling the intralanguage inheritance relations, and it inherits information from a shared part via interlanguage inheritance relations. In the parameterised model, on the other hand, all inheritance relations (inter- and intralanguage) are integrated into one single hierarchy and language parameters are used to indicate which parts of the hierarchy are valid for which language. Thus no distinction is being made between inter- and intralanguage inheritance relations.

As a result of the fact that all information is integrated into a single hierarchy, there is less repetition of inheritance relations in a parameterised model than in a non-parameterised model. Every relation is specified only once in a parameterised

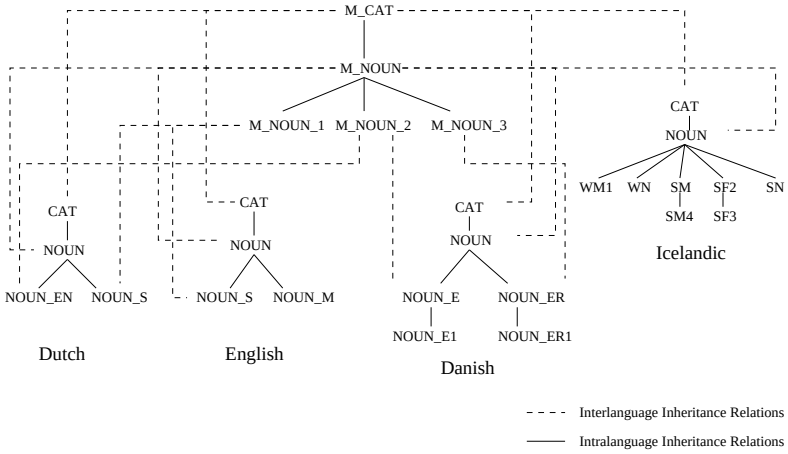


Figure 7.1: Extract of the actual noun class hierarchy in the SS/BODYPART implementation

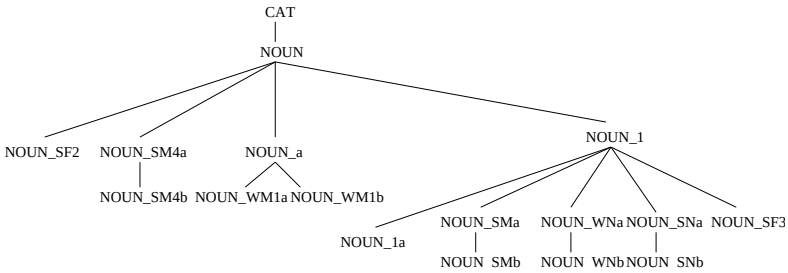


Figure 7.2: Extract of the actual noun class hierarchy in the META/BODYPART implementation

model, whereas there is repetition of the inheritance relations at the mono- and multilingual levels in a non-parameterised model.

The way that inter- and intralanguage inheritance relations are expressed also affects the definition of direct interlanguage inheritance where one language inherits characteristics directly from another language such as in the case of borrowings. For example, Dutch borrowed the word *computer* from English and the Dutch lexeme for *Computer* could inherit its phonological form directly from English using interlanguage inheritance relations from English to Dutch.

In the non-parameterised Structure-Sharing model, languages inherit shared information from shared hierarchies, and as there is no language feature, there is, in principle, nothing to indicate which hierarchy represents which language. Consequently, modelling direct interlanguage inheritance is not straightforward in this model. In theory, it is possible for the monolingual lexicons to inherit information directly from each other without going via a shared hierarchy, but the actual implementation of a lexicon allowing such inheritance relations is complicated by several engineering issues. Incorporating direct inheritance relations means that the monolingual lexicons are not completely separate anymore. For this to work, one has to make sure that there are no overlapping node names in the different language-specific lexicons, for example, by introducing language identifiers in the node names. Another side-effect of allowing direct interlanguage inheritance relations is that the resulting multilingual inheritance network becomes messier. There are now several inheritance routes possible for expressing the same shared phenomenon. There is no uniform treatment of interlanguage inheritance anymore.

Direct interlanguage inheritance relations can generally be expressed more easily in a parameterised model because all information is integrated into a single hierarchy. In this architecture, a Dutch lexeme could, for example, inherit information directly from English as easily as from Germanic in general.

By allowing direct interlanguage inheritance in the lexicon, a distinction can be made between the different kinds of similarities that are found between languages. We saw earlier that languages may exhibit similarities for different reasons, e.g. genetic origin, language contact or chance. It seems reasonable, from a linguistic perspective, to specify similarities based on genetic origin in a common part and those due to language contact by means of direct interlanguage inheritance links. This brings us to the second issue, multilingual information sharing.

7.1.2 Multilingual Information Sharing

This issue concerns which information can and should be shared in a multilingual inheritance lexicon, how it is shared, and whether the choice of model has an effect on this. The analysis and implementation of the sample fragments shows that the latter is the case. The Micro-Features model was discarded early on in the thesis, because it cannot capture the kind of cross-linguistic generalisations that natural languages exhibit. It assumes that all information is shared between languages except for some local, low-level phenomena. For the other models, sample fragments were implemented as described in Chapter 6. They all cover the same data, but differ in the way that generalisations can be captured.

The Infinitesimal model and the MetaTheory model are the most flexible. They can capture cross-linguistic generalisations into a single definition at each level in the implementation, i.e. lexeme, syllable sequence, syllable, and phoneme. The Meta-Features model only allows one to do so at the highest level, i.e. the lexeme level, but not lower down. In the Structure-Sharing model, it is not possible at all to capture cross-linguistic information into a single definition as there is no language feature in this model. It should be noted though that more cross-linguistic research is required to determine at which levels in the implementation, we really would like to capture generalisations. We return to this issue in our discussion of future work.

Let us now look at the amount of information that can be shared. As noted in Bateman et al. (1999), the amount of information that can be shared between languages does not only depend on the closeness of the languages.

> Even within traditional sets of typologically related languages (e.g., English, German, French, Dutch) some languages will be more closely related than others, and moreover, the closeness of the relationship will vary *depending on what part of the linguistic system one is examining*. (Bateman, Matthiessen, and Zeng, 1999, p. 623)

Table 7.1 shows the amount of shared information at the phoneme, syllable, and syllable sequence levels in our implementation of the SS/MED fragment. The percentages at the different levels are based on the number of shared phonemes, syllables and syllable sequences.

According to this table, the largest amount of information sharing is found at the

	D,E,DK	D, DK	DK, E	E, D
phoneme level	37%	46%	46%	46%
syllable level	2%	7%	3%	6%
syllable sequence level	61%	71%	67%	68%

Table 7.1: Information sharing based on SS/MED

syllable sequence level. 61% of all lexical entries in our medical test data consists of the same number of syllables in all three languages.

There is very little sharing at the syllable level. This is, however, consistent with data in the CELEX (Baayen, Piepenbrock, and van Rijn, 1995) database. Calculations on the CELEX database show that there is about 7% sharing of syllables between Dutch and English.

At the lexeme level, we compared how many lexical rules and syntactic categories and subcategories are distinguished in the different languages and how many of those are specified in the common part. English seems to be the default at this level, with most of its lexical rules and syntactic categories being specified in the shared part.

7.1.3 Development Strategy

We saw in Chapter 6 that the choice of development strategy depends on the amount of integration in the lexicon. Models which build a less integrated lexicon are more flexible with respect to development strategy than models which construct more integrated lexicons. From the models investigated the Structure-Sharing model is the most flexible and allows both a parallel and non-parallel development of the multilingual lexicon. The parameterised models, on the other hand, clearly prefer a parallel development strategy. The more cross-linguistic knowledge is available from the start, the more reliable the multilingual inheritance hierarchy that is constructed. This is important since everything depends on this hierarchy and once it has been defined, it is difficult to change. Thus, the 'correctness' of the multilingual hierarchy determines the extendability and reusability of a parameterised model.

The extendability and reusability of the Structure-Sharing model does not solely depend on the information contained in the shared hierarchy. A new language

is added by defining a language-specific hierarchy plus a set of links from this hierarchy to the shared hierarchy. There is, however, no guarantee that there will be optimal sharing.

7.1.4 Further contributions

The thesis provided a testbed for the lexical description framework developed by Tiberius and Evans (2000) which was described in Chapter 5. This framework was used for the implementation of the sample fragments. It can be seen as a development of the framework used in the PolyLex project, extending PolyLex's word model down to the level of phonological features and adopting a more modular and more uniform phonologically-based approach to lexical generalisation. This way, the framework provides a flexible means of capturing lexical generalisations within and across languages.

This framework was used to capture morphological, phonological, and morphophonological similarities between languages, and the insights gained in this respect contribute to the development of a theory for the multilingual lexical representation of phonology.

Finally, the thesis explored the concept of metaphonemes as introduced in Tiberius and Cahill (2000a; 2000b). In the context of this thesis, cross-linguistic phoneme correspondences were defined for the vowel phonemes in Dutch, English, and Danish and incorporated in the sample fragments. This work is being continued in the METAPHON project[1].

7.2 Future development

Multilingual lexical representation is a fairly new area of research and this thesis can be no more than an interim exploration. This section suggests several avenues for further research.

[1] http://www.itri.brighton.ac.uk/projects/metaphon/

7.2.1 Further exploration of the models

In Chapter 6, we saw that the implementation of the MetaTheory model was complicated by theoretical issues. There is nothing in the model which tells us what the metatheory for a set of languages should look like. That is, should inheritance relations be captured at the metatheory level, at the object theory level or at both, and how should these inheritance relations be structured? These issues need to be resolved first before a proper implementation of the model can be attempted.

In this thesis, we explored five models separately. In future work, we might like to explore combinations of these models. For example, although the Micro-Features model is not suitable for NLP on its own, in combination with a model such as the Structure-Sharing model, it provides a neat way of capturing cross-linguistic generalisations at the phoneme level. Furthermore, new models might emerge.

7.2.2 Multilingual Information Sharing

The sample implementations discussed in this thesis, should mainly be considered from an NLP perspective. Sharing decisions have generally been made on pure 'mathematical' grounds. That is, information which is common to the majority of the languages in the lexicon is shared. One reason for adopting this 'data-driven' approach to information sharing is that a more linguistically justified model will require a better understanding of the amount of variation that natural languages exhibit and at which levels they exhibit variation. This kind of information will allow us to distinguish true universals from spurious ones which will guide us when building a multilingual inheritance lexicon. It may, for example, provide an insight into how restricted the Infinitesimal model should be.

The language tree (although very basic) that is used in the parameterised models, is currently based on genetic relationships between languages. However, we saw in Chapter 4 that other factors such as typological relations and language contact may also play a role. These factors may also have to be taken into account when constructing the language hierarchy and this may affect the way information is shared in the parameterised models.

	English	Dutch	Danish	Shared
Diameter	d a I	d i:	d i	d
	{	j a:	a	
	m I	m e:	m e:	m e:
	t @ r	t @ r	d O_o	t @ r

Table 7.2: Phonemic transcriptions for the lexical entry *Diameter*

7.2.3 The status of the syllable in multilingual lexical representation

In this thesis, we have investigated different approaches to the sharing of information in the syllable module which were discussed in Section 6.2. None of them seems ideal.

- **Cross-linguistic sharing of identical segments in identical syllables in identical words**

  In this approach, identical segments are shared if they occur in the same syllable in the same word in most languages of the sample. Stress has not been taken into account. For clarity, this approach is illustrated here with the lexeme *Diameter* in Table 7.2. The fifth column represents the shared information. Thus the multilingual lexical entry for *Diameter* consists of four syllables. The first syllable consists of a shared `/d/` followed by a vowel. No element is shared in the second syllable. The third syllable is `/me:/` as shared by Dutch and Danish, and the fourth syllable is `/t@r/` as shared by English and Dutch.

  This approach is problematic in a few respects. First, identical syllables can occur in different languages, but not necessarily in the same words and in the same position. For example, the syllable `/me:/` occurs in both Dutch and Danish. Sometimes it can be shared, sometimes it cannot. However, sharing the same syllable in one lexical entry, but not in another, suggests that we

have a different syllable in the different lexeme entries, i.e. the syllable cannot be collapsed into an abstract syllable in the common syllable inventory. It also means that there is a lot of redundancy in the syllable inventories. Second, there is a practical disadvantage, the syllables in the multilingual inventory are not necessarily real syllables. They are an abstraction of what is shared at the syllable level cross-linguistically and some only consist of the shared values for onsets, peaks, and/or codas, such as the first syllable of *Diameter*. Consequently, the multilingual syllable inventory can get quite messy and difficult to understand.

- **Cross-linguistic sharing of segmentally identical syllables**
  In this approach, segmentally identical syllables which occur in the majority of the languages of the sample are shared regardless of the words in which they occur, their position (i.e. 1^{st}, 2^{nd}, 3^{rd} syllable) in those words, and regardless of stress. Thus, the syllable `/me:/` which occurs in both Dutch and Danish will be included in a multilingual syllable inventory. This approach has the advantage that all syllables in the multilingual syllable inventory are real syllables, they are not abstractions of actually occurring syllables. This results in a fairly neat and perspicuous multilingual syllable inventory. The disadvantage is that it is questionable whether it really results in optimal sharing. A comparison of the syllables in the CELEX database shows that only 1646 syllables out of 21966 are shared between Dutch and English, which is 7.49%. Including syllable position these figures are 3165 out of 50532, which is 6.26%. This does not seem a lot.

In both approaches the amount of shared information can be topped up by capturing identical syllables including metaphonemes, such as the |{A| correspondence in the lexical entry for *Hand* in Dutch `/hAnt/` and English `/h{nd/`.

Problematic for both approaches are those cases where there is not a one-to-one correspondence between the syllables in the different languages. We illustrated this in Chapter 6 with the lexeme *Elbow* which is a disyllabic word in English `/El-b@U/` and a trisyllabic word in Dutch `/E-l@-bo:x/`. See Table 6.3. Ideally, what we would like to have is a multilingual syllabification theory which does not enforce strict syllable boundaries at the multilingual level and which allows the insertion and deletion of elements as required taking language-specific stress into account.

7.2.4 Multilingual lexical representation and semantics

The sample lexicons in this thesis focus on the sharing of morphological, phonological and morphophonological information and no attention has been paid to semantics. Although the lexical entries were chosen such that they are translation equivalents, their actual meaning has not been encoded in the lexicons.

This deficiency can be overcome. The lexical description framework that we use, organises the lexicon into distinct self-contained modules corresponding to the different levels of lexical description, and a separate module could be added for the semantics as is shown in Figure 7.3. Here, the Dutch lexeme *Gebed* ('prayer') inherits its semantics from `PRAY` in the semantic module.

In Chapter 2 we saw that the semantics of one language can be mapped onto that of another by constructing a conceptual hierarchy which is shared by all the languages in the lexicon. The lexical entries will then be mapped onto concepts in the ontology by means of mapping rules. This approach would be consistent with the general spirit of our approach to multilingual lexical representation which is inheritance-based.

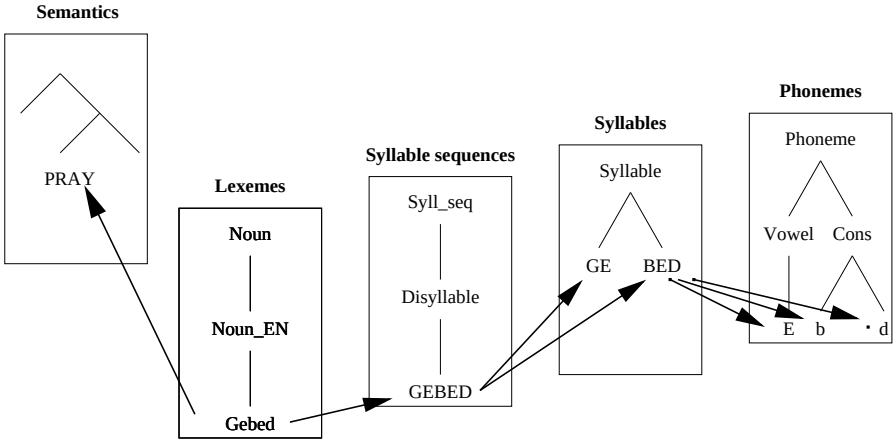


Figure 7.3: Multiple inheritance network with semantics

A small experiment along this line was done integrating parts of the Mikrokosmos ontology (Nirenburg et al., 1996) into a fragment of the Structure-Sharing body part lexicon.

7.2.5 Machine Learning and Multilingual Lexical Representation

In Chapter 6, we mentioned the possibility of using machine learning techniques to construct the multilingual noun hierarchy in the parameterised models. The use of machine learning techniques has not been explored in this thesis. However, it would be interesting to see whether such techniques could be used to construct the multilingual inheritance structure in the different models. A step towards partly automating the construction of a multilingual Structure-Sharing lexicon was undertaken by Cahill (1998). As we saw in Chapter 6, she describes a method for the automatic acquisition of data for the Structure-Sharing model after the initial manual setting up of the hierarchy.

7.2.6 Towards a multilingual featural description

In Chapter 5, we saw that the lexical description framework that we adopted in this thesis uses a single phonology-based representation going down to the level of phonological features to capture lexical generalisations traditionally modelled as phonology and morphology. However, in the implementation, we do not take full advantage of the featural description and limit ourselves to segmental phonology. Extending the level of representation to a featural level would permit greater economies of representation. A featural description would reduce the amount of redundancy in the definition of the lexeme entries. For example, at the moment, the values of onsets, peaks, and codas are inherited from nodes which bundle up the feature-values of the particular phonemes. These nodes may inherit from each other – for example, the nodes defining the phonemes /f/ and /v/ only differ in the value for the feature [voice] – but in the lexeme entry for *Finger*, the English onset will inherit its value from the node defining the phoneme /f/ to form /fINg@r/ and the Dutch onset will inherit its value from the node defining the phoneme /v/ to form /vIN@r/. If we would get rid of the segmental phoneme representations in the definition of the lexeme entries, generalisations such as between the phonemes /f/ and /v/ could be captured at the level of the lexeme entries. Furthermore, using featural descriptions at a segmental and supra-segmental level will also make the extension to speech processing possible as shown in Cahill (1993) and Carson-Berndsen (1998).

7.3 Summary and Conclusions

This thesis discussed a relatively new approach to multilingual lexical representation which moves away from the traditional Machine Translation architecture to multilingual lexicons. Rather than linking the monolingual lexicons at the level of semantics only, the aim is to encode and exploit lexical similarities between related languages at all levels of linguistic description – morphology, phonology, etc. – by using an inheritance-based formalism.

Our main goal was not to develop a multilingual lexicon for a practical NLP system, but to explore different methodological and theoretical issues involved in the development of multilingual inheritance-based lexicons. This thesis focussed in particular on three issues, viz. the regulation of the inter- and intralanguage inheritance relations, multilingual information sharing and development strategies. It explored these issues by comparing different architectures that can be used to construct a multilingual inheritance-based lexicon. Two kinds of architectures were distinguished, i.e parameterised and non-parameterised.

From the parameterised models, the Micro-Features model was discarded early on as it cannot capture the kind of generalisations that natural languages exhibit. The MetaTheory model is not a full competitor either as the full potential of this model can only be appreciated once we have developed a theory which tells us what the metatheory for a set of languages should look like.

The thesis has shown that at the moment, the question which of the remaining models provides the best way to build a multilingual inheritance-based lexicon is difficult to answer. It seems that which model is best, depends on what one wants to do with it.

For practical applications, the non-parameterised Structure-Sharing model seems currently the most suitable model. It is relatively straightforward to construct. Each monolingual lexicon keeps its own inheritance structure and shared information is specified in a shared hierarchy from which the monolingual lexicons inherit. There are no preconditions to its construction, i.e. it does not require that all data is available from the start. The disadvantage of the Structure-Sharing model is that there is a lot of redundancy in the model which may make the inheritance network quite messy especially when the network gets bigger. One will have to live with this.

The construction of a parameterised multilingual lexicon is less straightforward. Parameterised lexicons require more preparatory work. All cross-linguistic data has to be available from the start (which can be quite time-consuming) and in the more powerful models, such as the Infinitesimal model, one has to decide at which levels language variation is allowed in the multilingual lexicon. However, the state of the art in language typology and cross-linguistic research is in general not far enough advanced to guide us in making such decisions. Because of these difficulties, the parameterised models are currently less appealing for practical applications.

From a theoretical perspective, the parameterised models – and in particular the Infinitesimal model – are more interesting than the non-parameterised Structure-Sharing model. As the Infinitesimal model allows us to capture different kinds of generalisations in different ways, it is better placed to provide a linguistic model of the relationships that exist between languages than the other models. It may even provide a formal account of how languages have diverged from their common origin.

Appendix A

General Implementation Conventions

The sample fragments in this thesis have been implemented and tested running Sussex/Brighton DATR-2.8 under Sussex Poplog Prolog. Each sample fragment is delivered as a suite of DATR files in a single directory.

File Organisation

The files in these directories can be grouped together into two layers, a layer forming the surface lexicon and a layer forming the core infrastructure.

- **The surface lexicon**
  The files in this layer contain linguistic information. They define the individual lexical entry nodes using the lexical description framework which was described in Chapter 5.

- **The framework infrastructure**
  This layer defines the core (mainly non-linguistic) mechanisms of the framework such as the rule application mechanism and display routines. Properties such as feature structures and rule names are also declared here. Currently, this layer only contains one file.

File naming conventions

In the implementation, the following file naming conventions have been used. The file defining the framework infrastructure is called `infrastruc.dtr`. Within the

surface lexicon, the following distinctions are made. Files belonging to the data set of body part terms are called `pmed` files, files belonging to the implementation of the medical test set are called `med` files. Furthermore, specifications such as `lex`, `syll`, and `phon` are used in the file names to refer to the different kinds of linguistic information that they contain. Files with the specification `lex` contain lexical entry definitions, whereas `syll` files define syllable inventories and `phon` files define phoneme inventories. Sometimes, the specification `lexphon` is also used. In that case, the morphological and phonological specification of the lexical entries are defined in two separate files. The `lexphon` files contain the phonological definition of the lexemes. The morphological information of the lexical entries is still defined in the `lex` files.

In addition, language identifiers are used in the implementation of the Structure-Sharing model to refer to files which contain language-specific information. The name of the language is added at the beginning of the file name. Files containing shared information are prefixed with `multi` in the Structure-Sharing Model.

Node naming conventions

Nodes have been given mnemonic names to facilitate understanding. This has no significance for the operation of the system, but it makes the lexicons easier to understand and maintain.

The general DATR convention is that node names start with an uppercase letter, attributes and values do not. No deviations have been made from this scheme. The node naming conventions used in the sample fragments reflect the modular structure of our lexical description framework and is illustrated in figure A.1.

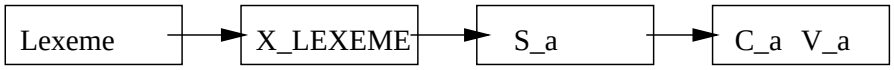


Figure A.1: Lexical Description Framework with illustration of node naming conventions

Lexeme nodes start with an uppercase letter followed by lowercase letters – `Lexeme`. Nodes defining the phonological form of lexemes are all in uppercase and (often)

prefixed by X_ – X_LEXEME. Nodes which define syllables are prefixed with S_ followed by the actual syllable in SAM-PA-based transcription[1]. The names for nodes defining phonemes are prefixed either with C_ for consonants or V_ for vowels. Thus, C_l defines the phoneme /l/ and V_@ defines the phoneme /@/.

In addition, the following conventions have been used. Nodes defining instances of suffixes start with Suffix_ followed by the actual value of the suffix. Nodes which define language-specific information only are generally affixed with a language abbreviation in uppercase, i.e. D for Dutch, E for English, DK for Danish and I for Icelandic. For example, X_FOOT/D is the name of a node defining the phonological form for *foot* in Dutch. The names for nodes which define lexical rules are all in uppercase, e.g. SING_NOUN_RULE contains the definition of the singular formation of nouns. Nodes defining syntactic categories have names written in uppercase. The names of nodes which define the different components that make up a lexeme, i.e. suffix, syllable sequence, syllable, are also all in uppercase. Finally, there is a set of nodes which define special functions. For example, the node LASTSYLL finds the last syllable of a lexeme. The names of these nodes are all in uppercase.

[1]The transcription differs from SAM-PA conventions in that long vowels are represented by duplicating the vowel rather than using the length marker :. Thus for example the node S_saal defines the syllable /sa:l/.

Appendix B

Test Data

This appendix contains the test sets that are used in the sample lexical fragments. Two data sets were compiled. A small data set consisting of 19 body part terms (all nouns) in Dutch, Danish, English, and Icelandic, and a larger data set of 100 medical terms (nouns and adjectives) taken from the online Multilingual Glossary of technical and popular medical terms in nine European languages (1995) in Dutch, Danish, and English and partly in Icelandic. The transcriptions were extracted from CELEX (Baayen, Piepenbrock, and van Rijn, 1995) for Dutch and English, from Hansen (1990) for Danish and from Blondäl (1920 1924) and Einarsson (1945) for Icelandic and use the SAM-PA phonetic alphabet (Wells, 1987; Wells, 1989; Wells, 1995).

B.1 Body Parts Test Set

E	: ankle	/{Nkl/
D	: enkel	/ENk@l
DK	: ankel	/ANg@l/
I	: ökkli	/9klI/
E	: arm	/A:m/
D	: arm	/Arm/
DK	: arm	/A:m/
I	: armur	/Armur/
E	: breast	/brEst/
D	: borst	/bOrst/
DK	: bryst	/bR2sd/
I	: brjóst	/brj@Ust/
E	: ear	/I@r/
D	: oor	/o:r/
DK	: øre	/2:_rO_o/
I	: eyra	/eIr/
E	: elbow	/Elb@U/
D	: elleboog	/El@bo:x/
DK	: albue	/albu:@/
I	: olnbogi	/OlboijI/
E	: eye	/aI/
D	: oog	/o:x/
DK	: øje	/O_oj@/
I	: auga	/9IGA/
E	: finger	/fINg@r/
D	: vinger	/vIN@r/
DK	: finger	/feNO_o/
I	: fingur	/fINgur/

E : foot /fUt/
D : voet /vu:t/
DK : fod /foD/
I : fótur /f@Utur/

E : hair /hE@r/
D : haar /ha:r/
DK : hår /hQ:/
I : hár /haUr/

E : hand /h{nd/
D : hand /hAnt/
DK : hånd /hO_on/
I :

E : head /hEd/
D : hoofd /ho:ft/
DK : hoved /ho:@D/
I : höfuð /h9vYT/

E : heel /hiil/
D : hiel /hi:l/
DK : hæl /hE:l/
I : hæll /haIdl/

E : mouth /maUT/
D : mond /mOnt/
DK : mund /mOn/
I :

E : neck /nEk/
D : nek /nEk/
DK : nakke /nAg@/
I : háls /haUls/

E : nose /n@Uz/
D : neus /n2:z/
DK : næse /nE:s@/
I : nef /nEv/

E : shoulder /S@Uld@r/
D : schouder /sxAud@r/
DK : skulder /sgulO_o/
I : öxl /2gsl/

E : stomach /stVm@k/
D : maag /ma:x/
DK : mave /ma:u_^@/
I : magi /maIjI/

E : toe /t@U/
D : teen /te:n/
DK : tå /tO:/
I : tá /taU/

E : tooth /tuuT/
D : tand /tAnt/
DK : tand /tan/
I : tönn /t2n/

B.2 Medical Test Set

E :	aerosol	/E@-r@U-sQl/
D :	aërosol	/a:-e:-ro:-sOl/
DK :	aerosol	/E-Ro-so:l/
I :		

E :	alcoholism	/{l-k@-hQ-lI-z@m/
D :	alcoholisme	/Al-ko:-ho:-lIs-m@/
DK :	alkoholisme	/al-ko-ho-lis-m@/
I :		

E :	alert	/@-l3:t/
D :	alert	/a:-lErt/
DK :	kvik	/kvig/
I :		

E :	analysis	/@-n{-l@-sIs/
D :	analyse	/a:-na:-li:-z@/
DK :	analyse	/a-na-ly:-s@/
I :	greining	/grEI:-niNk/

E :	antibiogram	/{n-tI-baI-Q-gr{m/
D :	antibiogram	/An-ti:-bi:-jo:-GrAm/
DK :	antibiogram	/an-ti-bi-o:-gRAm/
I :		

E :	antibody	/{n-tI-bQ-dI/
D :	antilichaam	/An-ti:-lI-xa:m/
DK :	antistof	/an-ti-sdO_of/
I :	mótefni	/mout-Eb-nI/

E :	antidote	/{n-tI-d@Ut/
D :	antidotum	/An-ti:-do:-t}m/
DK :	antidotum	/an-ti-do:-tOm/
I :	móteitur	/mout-Ei:-d2r/
E :	antipsychotic	/{n-tI-saI-kQ-tIk/
D :	antipsychoticum	/An-ti:-psi:-xo:-ti:-k}m/
DK :	neuroleptikon	/n2u_^-Ro-lEb-ti-ka/
I :		
E :	aromatic	/{-r@U-m{-tIk/
D :	aromatisch	/a:-ro:-ma:-ti:s/
DK :	aromatisk	/A-Ro:-ma:-tisg/
I :		
E :	aura	/O:-r@/
D :	aura	/Au-ra:/
DK :	aura	/Au_^-RA/
I :		
E :	autonomic	/O:-tQ-n@-mIk/
D :	autonoom	/Au-to:-no:m/
DK :	autonom	/Au_^-to-no:-m@/
I :		
E :	biochemical	/baI-@U-kE-mI-kl/
D :	biochemisch	/bi:-jo:-xe:-mi:s/
DK :	biokemisk	/bi:-o-ke:-misg/
I :		
E :	biological	/baI-@U-lQ-dZI-kl/
D :	biologisch	/bi:-jo:-lo:-Gi:s/
DK :	biologisk	/bi-o-lo:-u_^isg/
I :		

E :	biosynthesis	/baI-@U-sIn-T@-sIs/
D :	biosynthese	/bi:-jo:-sIn-te:-z@/
DK :	biosyntese	/bi-o-syn-te:-s@/
I :		

E :	circulation	/s3:-kjU-leI-Sn,/
D :	circulatie	/sIr-ky:-la:-tsi:/
DK :	cirkulation	/siR-ku-la-S'o:?n/
I :		

E :	classic	/kl{-sIk/
D :	klassiek	/klA-si:k/
DK :	klassisk	/kl'a-sisg/
I :		

E :	collapse	/k@-l{ps/
D :	collaps	/kO-lEps/
DK :	kollaps	/kO_o-l'Abs/
I :		

E :	contact	/kQn-t{kt/
D :	contact	/kOn-tAkt/
DK :	kontakt	/kO_on-t'Agd/
I :		

E :	contra-indication	/kQn-tr@-In-dI-keI-Sn,/
D :	contra-indicatie	/kOn-tra:-In-di:-ka:-tsi:/
DK :	kontraindikation	/k'O_on-tRA-en-di-ka-S'o:?n/
I :	frábending	/frAu:-bEn-diNk/

E :	curve	/k3:v/
D :	curve	/k}r-v@/
DK :	kurve	/k'uR-u_^@/
I :	hnít	/hnI:t/

E :	cycle	/saI-kl,/
D :	cyclus	/si:-kl}s/
DK :	cyklus	/s'y-klus/
I :		

E :	cyclic	/saI-klIk/
D :	cyclisch	/si:-kli:s/
DK :	cyklisk	/s'y-glisg/
I :		

E :	diagnosis	/daI-@g-n@U-sIs/
D :	diagnose	/di:-ja:-Gno:-z@/
DK :	diagnose	/di-ag-n'o:-s@
I :	sjúkdómsgreining	/sju:k-doums-grEi:-niNk/

E :	diameter	/daI-{-mI-t@r/
D :	diameter	/di:-ja:-me:-t@r/
DK :	diameter	/di-a-m'e:?-dO_o/
I :	þvermál	/TvEr-mAul/

E :	diarrhoea	/daI-@-rI@/
D :	diarree	/di:-jA-re:/
DK :	diarré	/di-A-R'E:?/
I :		

E :	differentiation	/dI-f@-rEn-SI-eI-Sn,/
D :	differentiatie	/dI-f@-rEn-Sa:-tsi:/
DK :	differentiering	/di-f@-R{n-S'e:?-ReN/
I :	aðgreining	/aD-grEi-niNk/

E :	diffusion	/dI-fju:-Zn,/
D :	diffusie	/dI-fy:-zi:/
DK :	diffusion	/di-fu:-So:n/
I :		

E : dose /d@Us/
D : dosis /do:-z@s/
DK : dosis /do:-sis/
I :

E : eczema /Ek-sI-m@/
D : eczeem /Ek-se:m/
DK : eksem /Eg-s'e:?m/
I : þref /TrE:v/

E : effect /I-fEkt/
D : effect /E-fEkt/
DK : effekt /E-f'Egd/
I : verkun /vEr_0-g2n/

E : efficient /I-fI-S@nt/
D : efficient /E-fi:-SEnt/
DK : effektiv /'E-fEg-t,iu_^?/
I :

E : erection /I-rEk-Sn,/
D : erectie /e:-rEk-si:/
DK : erektion /e-REg-S'o:?n/
I :

E : fraction /fr{k-Sn,/
D : fractie /frAk-si:/
DK : fraktion /fRAg-S'o:?n/
I : brot /bro:t/

E : frequency /fri:-kw@n-sI/
D : frequentie /fr@-kwEn-si:/
DK : frekvens /fRE-kv'En?s/
I :

E :	glucose	/glu:-k@Us/
D :	glucose	/xly:-ko:-z@/
DK :	glukose	/gly-k'o:-s@/
I :		

E :	gynaecological	/gaI-n@-k@-lQ-dZI-kl,/
D :	gynaecologisch	/xi:-ne:-ko:-lo:-Gi:s/
DK :	gynækologisk	/gy-nE-ko-l'o:?-Wisg/
I :		

E :	hydration	/haI-dreI-Sn,/
D :	hydratatie	/hi:-dra:-ta:-tsi:/
DK :	hydrering	/hy-dRe:-ReN/
I :	vötnun	/v9hd-n2n/

E :	hydrofobic	/haI-dr@-f@U-bIk/
D :	hydrofoob	/hi:-dro:-fo:b/
DK :	hydrofob	/hy-dRo-fob/
I :		

E :	hypothermia	/haI-p@U-T3:-mI@/
D :	hypothermie	/hi:-po:-tEr-mi:/
DK :	hypotermi	/hy-po-t{R-m'i:/
I :		

E :	identification	/aI-dEn-tI-fI-keI-Sn,/
D :	identificatie	/i:-dEn-ti:-fi:-ka:-tsi:/
DK :	identifikation	/i-dEn-ti-fi-ka-S'o:?n/
I :		

E :	immunological	/I-mju:-n@-lQ-dZI-kl,/
D :	immunologisch	/I-my:-no:-lo:-Gi:s/
DK :	immunologisk	/i-m'u:?-no-l'o:?-Wisg/
I :		

E :	indication	/In-dI-keI-Sn,/
D :	indicatie	/In-di:-ka:-tsi:/
DK :	indikation	/en-di-ka-S'o:?n/
I :		

E :	inhalation	/In-h@-leI-Sn,/
D :	inhalatie	/In-ha:-la:-tsi:/
DK :	inhalation	/en-ha-lA-S'o:n/
I :	innöndun	/In-9n-d2n/

E :	initial	/I-nI-Sl,/
D :	initiaal	/i:-ni:-tSa:l/
DK :	initial	/i-ni-ti'-a:?l/
I :		

E :	inspiration	/In-sp@-reI-Sn,/
D :	inspiratie	/In-spi:-ra:-tsi:/
DK :	inspiration	/en-sbi-RA-S'o:?n/
I :	innöndun	/In-9n-d2n/

E :	intelligence	/In-tE-lI-dZ@ns/
D :	intelligentie	/In-t@-li:-GEn-si:/
DK :	intelligens	/en-tE-li-g'En?s/
I :	greind	/grEint/

E :	interindividual	/In-t3:r-In-dI-vI-dZU@l/
D :	interindividueel	/In-t@r-In-di:-vi:-dy:-we:l/
DK :	interindividuel	/en-dO_o-en-di-vi-du-El/
I :		

E :	interval	/In-t@-vl,/
D :	interval	/In-t@r-vAl/
DK :	interval	/en-dO_o-v'al?/
I :		

E :	intolerance	/In-tQ-l@-r@ns/
D :	intolerantie	/In-to:-l@-rAn-si:/
DK :	intolerance	/'en-tO-l@-R,AN-s@/
I :		

E :	lytic	/lI-tIk/
D :	lytisch	/li:-ti:s/
DK :	lytisk	/ly-tisg/
I :		

E :	mania	/meI-nj@/
D :	manie	/ma:-ni:/
DK :	mania	/m'a-nja/
I :	æði	/ai:-DI/

E :	massage	/m{-sA:Z/
D :	massage	/mA-sa:-Z@/
DK :	massage	/ma-s'a:-S@/
I :	nudd	/nYt/

E :	medicinal	/m@-dI-sI-nl,/
D :	medicinaal	/me:-di:-si:-na:l/
DK :	medicinsk	/me-di-s'i:?nsg/
I :		

E :	microcirculation	/maI-kr@U-s3:-kjU-leI-Sn,/
D :	microcirculatie	/mi:-kro:-sIr-ky:-la:-tsi:/
DK :	mikrocirkulation	/m'i-kRo-siR-ku-la-S'o:?n/
I :		

E :	minimum	/mI-nI-m@m/
D :	minimum	/mi:-ni:-m}m/
DK :	minimum	/m'i:?-ni-mOm/
I :		

E : mobility /m@U-bI-l@-tI/
D : mobiliteit /mo:-bi:-li:-tEIt/
DK : mobilitet /mo-bi-li-t'e:?d/
I :

E : musculature /mV-skjU-l@-tjU@r/
D : musculatuur /m}s-ky:-la:-ty:r/
DK : muskulatur /mus-gu-la-t'uR?/
I :

E : nervousness /n3:-v@s-nIs/
D : nervositeit /nEr-vo:-zi:-tEIt/
DK : nervøsitet /n{R-v2_r-si-t'e:?d/
I :

E : neuropathy /njU@-rQ-p@-TI/
D : neuropathie /n—:-ro:-pa:-ti:/
DK : neuropati /n2u_^-Ro-pa-t'i:?/
I : taugakvilli /t9y-Ga-kvId-lI/

E : non-specific /nQn-sp@-sI-fIk/
D : aspecifiek /a:-spe:-si:-fi:k/
DK : uspecifik /u-sbe-si-fig/
I :

E : occasional /@-keI-Z@-nl,/
D : occasioneel /O-ka:-Zo:-ne:l/
DK : lejlighedsvis /l'AJ-li-heDs-v,i:?s/
I :

E : paediatric /pi:-dI-{-trIk/
D : pediatrisch /pe:-di:-ja:-tri:s/
DK : pædiatrisk /pE-di-a:-tRisg/
I :

E :	paranoia	/p{-r@-nOI-@/
D :	paranoia	/pa:-ra:-no:-ja:/
DK :	paranoia	/pAA-n'O_o-Ja/
I :		

E :	parasitic	/p{-r@-sI-tIk/
D :	parasitair	/pa:-ra:-si:-tE:r/
DK :	parasitisk	/pA:-si-tisg/
I :		

E :	passage	/p{-sIdZ/
D :	passage	/pA-sa:-Z@/
DK :	passage	/pa-s'a:-S@/
I :		

E :	perforation	/p3:-f@-reI-Sn,/
D :	perforatie	/pEr-fo:-ra:-tsi:/
DK :	perforation	/p{R-fo-RA-So:n/
I :	rauf	/r9y:v/

E :	pessary	/pE-s@-rI/
D :	pessarium	/pE-sa:-ri:-j}m/
DK :	pessar	/pe-s'A:?/
I :		

E :	pharmacon	/fA:-m@-kQn/
D :	farmacon	/fAr-ma:-kOn/
DK :	farmaka (farmaka=plural)	/f'A:?-ma-ka/
I :		

E :	phenomenon	/f@-nQ-mI-n@n/
D :	fenomeen	/fe:-no:-me:n/
DK :	fænomen	/fE-no-m'e:?n/
I :		

E :	phobia	/f@U-bj@/
D :	fobie	/fo:-bi:/
DK :	fobi	/fo-b'i:?/
I :		

E :	physical	/fI-zI-kl,/
D :	fysisch	/fi:-zi:s/
DK :	fysisk	/f'y:?-sisg/
I :		

E :	plexus	/plEk-s@s/
D :	plexus	/plEk-s}s/
DK :	plexus	/plEg-susk/
I :		

E :	polytherapy	/pQ-lI-TE-r@-pI/
D :	polytherapie	/po:-li:-te:-ra:-pi:/
DK :	polyterapi	/p'o-ly-te-RA-p'i:?/
I :		

E :	potency	/p@U-t@n-sI/
D :	potentie	/po:-tEn-si:/
DK :	potentia	/po-tEn-s@/
I :		

E :	psychiatric	/saI-kI-{-trIk/
D :	psychiatrisch	/psi:-xi:-ja:-tri:s/
DK :	psykiatrisk	/sy-ki'-a:?-tRisg/
I :		

E :	psychotropic	/saI-k@U-trQ-pIk/
D :	psychotroop	/psi:-xo:-tro:p/
DK :	psykofarmaka	/sy-ko-f'A:?-ma-ka/
I :		

E :	radiography	/reI-dI-Q-gr@-fI/
D :	radiografie	/ra:-di:-jo:-Gra:-fi:/
DK :	radiografi	//R'A:-di-o-gRA-f'i:?/
I :		

E :	radiotherapy	/reI-dI-@U-TE-r@-pI/
D :	radiotherapie	/ra:-di:-jo:-te:-ra:-pi:/
DK :	radioterapi	/R'A:-di-o-te-RA-p'i:?/
I :		

E :	reactivation	/rI-{k-tI-veI-Sn,/
D :	reactivering	/re:-Ak-ti:-ve:-rIN/
DK :	reaktivering	/RE-Ag-ti-ve:-O_o/
I :		

E :	recuperation	/rI-ku:-p@-reI-Sn,/
D :	recuperatie	/re:-ky:-p@-ra:-tsi:/
DK :	rekonvalescens	/RE-kO_on-va-l@s'Ens/
I :		

E :	reference	/rE-fr@ns/
D :	referentie	/re:-f@-rEn-si:/
DK :	reference	/REf@R'ANs@/
I :		

E :	reflective	/rI-flEk-tIv/
D :	reflectorisch	/re:-flEk-to:-ri:s/
DK :	reflekterende	/ RE-flEg-t'e:?-O_o-D@/
I :		

E :	regional	/ri:-dZ@-nl,/
D :	regionaal	/re:-Gi:-jo:-na:l/
DK :	regional	/RE-gi-o-n'a:?l/
I :		

E :	regulation	/rE-gjU-leI-Sn,/
D :	regulatie	/re:-Gy:-la:-tsi:/
DK :	regulering	/RE-gu-l'e:?-ReN/
I :	stjórnun	/sdj@Ur-dn2n/
E :	research	/rI-s3:tS/
D :	research	/ri:-s}rtS/
DK :	forskning	/f'Q:-sgneN/
I :		
E :	response	/rI-spQns/
D :	respons	/rEs-pOns/
DK :	respons	/RE-sb'O_on?s/
I :	svörun	/sv9r-2m/
E :	sedentary	/sE-dn,-t@-rI/
D :	sedentair	/se:-dEn-tE:r/
DK :	inaktiv	/en-Ag-ti_^/
I :		
E :	segment	/sEg-m@nt/
D :	segment	/sEG-mEnt/
DK :	segment	/seg-m'En?d/
I :		
E :	solution	/s@-lu:-Sn,/
D :	solutie	/so:-ly:-tsi:/
DK :	solutio(n)	/so-lu-S'o:?n/
I :	lausn	/l9ysn_0/
E :	sterilization	/stE-r@-laI-zeI-Sn,/
D :	sterilisatie	/ste:-ri:-li:-za:-tsi:/
DK :	sterilisering	/sde-Ri-li-s'e:-ReN/
I :	vönun	/v9-n2n/

E :	subjective	/s@b-dZEk-tIv/
D :	subjectief	/s}b-jEk-ti:f/
DK :	subjektiv	/s'ub-jEg-t,iu_^?/
I :		

E :	suspension	/s@-spEn-Sn,/
D :	suspensie	/s}s-pEn-zi:/
DK :	suspension	/sus-pEn-S'o:?n/
I :	lyfting	/lIf-diNk/

E :	superinfection	/su:-p@r-In-fEk-Sn,/
D :	superinfectie	/sy:-p@r-In-fEk-si:/
DK :	superinfektion	/s'u:?-bO_o-en-fEg-S'o:?n/
I :		

E :	symptom	/sImp-t@m/
D :	symptoom	/sImp-to:m/
DK :	symptom	/sym-t'o:?m
I :		

E :	thermoregulation	/T3:-m@U-rE-gjU-leI-Sn,/
D :	thermoregulatie	/tEr-mo:-re:-Gy:-la:-tsi:/
DK :	termoregulation	/t'{Rmo-RE-gu-la-S,o:?n/
I :	hitatemprun	/hI:-da-tEm_0-br2n/

E :	theoretical	/TI@-rE-tI-kl,/
D :	theoretisch	/te:-jo:-re:-ti:s/
DK :	teoretisk	/te-o-R'E:?-tisg/
I :		

E :	tone	/t@Un/
D :	tonus	/to:-n}s/
DK :	tone	/t'o:-n@/
I :	tónn	/t@Udn_0/

E :	tonic	/tQ-nIk/
D :	tonisch	/to:-ni:s/
DK :	tonisk	/to:-nisg/
I :		

E :	vegetative	/vE-dZI-teI-tIv/
D :	vegetatief	/ve:-G@-ta:-ti:f/
DK :	vegetative	/v'e-g@-ta-t,i-u_^?@/
I :		

E :	ventilation	/vEn-tI-leI-Sn,/
D :	ventilatie	/vEn-ti:-la:-tsi:/
DK :	ventilation	/vEn-ti-la-S'o:?n/
I :		

Bibliography

Ahmad, K., S. Hook, L. Lemnitzer, N. Moniano, J. Odijk, W. Paprotté, and F. Schumacher. 1993. MLEX$_d$ Standards for a Multifunctional Lexicon. Technical Report, CAP GEMINI INNOVATION for the MULTILEX Consortium, Paris. Final Report.

Allegranza, V., S. Krauwer, and E. Steiner, editors. 1991. *Machine Translation*, volume 6. Eurotra Special Issue.

Alshawi, H. (ed.). 1992. *The Core Language Engine*. The MIT Press, Cambridge, Massachusetts.

Anderson, S.R. 1988. Morphological Theory. In F.J. Newmeyer, editor, *Linguistics: the Cambridge survey*. Cambridge University Press, pages 146–191.

Antoni-Lay, M.H., G. Francopoulo, and L. Zaysser. 1994. A Generic Model for Reusable Lexicons: The Genelex Project. *Literary and Linguistic Computing*, 9(1):47–54.

Arnold, D., L. Balkan, R.L. Humphreys, S. Meijer, and L. Sadler. 1994. *Machine Translation: An Introductory Guide*. NCC Blackwell Ltd, Oxford.

Baayen, H., R. Piepenbrock, and H. van Rijn. 1995. The CELEX Lexical Database. Release 2 (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Barg, P. 1994. Automatic Acquisition of DATR Theories from Observations. Technical Report 59, Arbeiten des Sonderforschungbereichs 282.

Barg, P. 1996. Automatic Inference of DATR Theories. In H-H. Bock and W. Polasek, editors, *Data Analysis and Information Systems: Statistical and conceptual approaches (Proceedings of the 19th Annual Conference of the Gesellschaft Klassification)*, pages 506–515.

Bateman, J.A. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15–55.

Bateman, J.A., C. Matthiessen, and L. Zeng. 1999. Multilingual Natural Language Generation for Multilingual Software: A Functional Linguistic Approach. *Applied Artificial Intelligence*, 13(6):607–639.

Beard, R. 1981. *The Indo-European Lexicon: A Full Synchronic Theory*, volume 44. North-Holland Linguistic Series, Amsterdam.

Beckwith, R., C. Fellbaum, D. Gross, and G.A. Miller. 1991. WordNet: A lexical database organized on psycholinguistic principles. In U. Zernik, editor, *Lexical acquisition: Exploiting on-line resources to build a lexicon*. NJ: Erlbaum, Hillsdale, pages 211–232.

Bel, N., F. Busa, N. Calzolari, E. Gola, A. Lenci, M. Monanchini, A Ogonowski, I. Peters, W. Peters, N. Ruimy, M. Villegas, and A. Zampolli. 2000. SIMPLE: A General Framework for the Development of Multilingual Lexicons. In M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, and G. Stainhaouer, editors, *Proceedings of the Second International Conference on Language Resources and Evaluation*, volume III, pages 1379–1384, Athens, Greece, 31 May - 2 June.

Bell, A. 1978. Language Samples. In Greenberg J.H., editor, *Universals of human language. Volume I: Method & Theory*. Stanford University Press, pages 123–156.

Bennett, W.S. and J. Slocum. 1985. The LRC machine translation system. *Computational Linguistics*, 11:111–121.

Bird, S. and E. Klein. 1990. Phonological events. *Journal of Linguistics*, 26:33–56.

Blöndal, S.B.B. 1920-1924. *Islandsk - Dansk Ordbog*. Reykjavik.

Briscoe, T. 1991. Lexical Issues in Natural Language Processing. In E. Klein and F. Veltman, editors, *Symposium on Natural Language and Speech (Esprit Conference 1991)*. Springer Verlag, pages 39–68.

Briscoe, T., A. Copestake, and A. Lascarides. 1995. Blocking. In P. Saint-Dizier and E. Viegas, editors, *Computational Lexical Semantics*. Cambridge University Press, Cambridge, pages 273–302.

Briscoe, T., V. de Paiva, and A. Copestake (eds.). 1993. *Inheritance, Defaults, and the Lexicon.* Studies in Natural Language Processing. Cambridge University Press, Cambridge.

Cahill, L. 1990. *Syllable-based morphology for natural language processing.* Ph.D. thesis, School of Cognitive Science and Computing Sciences, University of Sussex, Brighton. also available as technical report CSRP 181.

Cahill, L. 1993. Morphophonology in the lexicon. In *Proceedings of the Fifth European Conference on Computational Linguistics*, pages 87–96.

Cahill, L. 1998. Automatic extension of a hierarchical multilingual lexicon. In *Multilinguality in the lexicon II*, pages 16–23, Brighton. ECAI. Workshop at the 13th biennial European Conference on Artificial Intelligence.

Cahill, L., J. Carson-Berndsen, and G. Gazdar. 2000. Phonologically based lexical knowledge representation. In F. Van Eynde and D. Gibbon, editors, *Lexicon Development for Speech and Natural Language Processing.* Kluwer, Dordrecht, pages 77–114.

Cahill, L. and G. Gazdar. 1995. Multilingual Lexicons for Related Languages. In *Proceedings of the 2nd DTI Language Engineering Conference*, pages 169–176.

Cahill, L. and G. Gazdar. 1996. A lexical analysis of numeral expressions in three related languages. In *Proceedings of the AISB-96 Workshop on Multilinguality in the Lexicon*, pages 69–75.

Cahill, L. and G. Gazdar. 1997. The inflectional phonology of German adjectives, determiners and pronouns. *Linguistics*, 35:211–245.

Cahill, L. and G. Gazdar. 1999a. German noun inflection. *Linguistics*, 35:1–42.

Cahill, L. and G. Gazdar. 1999b. The POLYLEX architecture: multilingual lexicons for related languages. *Traitement Automatique des Langues*, 40(2):5–23.

Calder, J. 1989. Paradigmatic morphology. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–65, Manchester, England.

Calzolari, N. 1998. An overview of Written Language Resources in Europe: a few Reflections, Facts, and a Vision. In *Proceedings of the First International Conference on Language Resources and Evaluation*, volume 1, pages 217–224, Granada Spain.

Carson-Berndsen, J. 1998. *Time map phonology: finite state models and events logics in speech recognition.* Kluwer, Dordrecht.

Chandioux, J. 1976. MÉTÉO: un système opérationnel pour la traduction automatique des bulletins météorologiques destinés au grand public. *META*, (21):127–133.

Chevalier, M., J. Dansereau, and G. Poulin. 1978. TAUM-MÉTÉO: Description du système. Technical Report, TAUM, Université de Montréal, janvier. Publication interne.

Coleman, J. 1992. 'Synthesis by rule' without segments or rewrite rules. In G. Bailly, C. benoit, and T.R. Sawallis, editors, *Talking Machines: Theories, Models and Designs.* Elsevier, Amsterdam, pages 43–60.

Coleman, J., A. Dirksen, S. Hussain, and J. Waals. 1996. Multilingual phonological analysis and speech synthesis. In *Computational Phonology in Speech Technology: Second Meeting of the ACL Special Interest Group in Computational Phonology*, pages 67–72, Santa Cruz. Association for Computational Linguistics.

Comrie, B. 1989. *Language universals and linguistic typology.* Oxford: Basil Blackwell, 2nd edition.

Copestake, A., B. Jones, A. Sanfilippo, H. Rodriguez, P. Vossen, S. Montemagni, and E. Marinai. 1992. Multilingual Lexical Representation. In *ESPRIT BRA-3030 ACQUILEX Working Paper*, number 043. University of Cambridge Computer Laboratory.

Corbett, G.G. and N.M. Fraser. 1993. Network Morphology: A DATR account of Russian nominal inflection. *Journal of Linguistics*, (29):113–142.

Daelemans, W. and G. Gazdar (eds.). 1992. *Special Issues on inheritance*, volume 18.2 & 18.3. Computational Linguistics.

Daelemans, W., K. De Smedt, and G. Gazdar. 1992. Inheritance in Natural Language Processing. *Computational Linguistics*, 18(2):205–218.

Dryer, M.S. 1989. Large linguistic areas and language sampling. *Studies in Language*, 13(2):257–292.

Eagles. 1996a. EAGLES Subcategorization Standards. Report of the EAGLES Lexicon/Syntax Group. Technical Report, SHARP Laboratories of Europe, Oxford, UK.

Eagles. 1996b. EAGLES Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora: A Common Proposal and Applications to European Languages. Technical Report EAG-CLWG-MORPHSYN/R, EAGLES.

Eagles. 1999. EAGLES Preliminary Recommendations on Semantic Encoding Final Report. Technical Report, EAGLES.

Einarsson, S. 1945. *Icelandic: Grammar, texts, glossary.* The Johns Hopkins University Press, Baltimore.

Evans, R. 1996. Exploiting inheritance in multilingual lexicons. Paper presented at the AISB-96 Workshop on Multilinguality in the Lexicon, Brighton, also available URL: `http://www.itri.brighton.ac.uk/~Roger.Evans/papers/aisb96`.

Evans, R. and G. Gazdar. 1996. DATR: A Language for Lexical Knowledge Representation. *Computational Linguistics*, 22(2):167–216.

Evans, R., G. Gazdar, and L. Moser. 1993. Prioritised Multiple Inheritance in DATR. In T. Briscoe, V. de Paiva, and A. Copestake, editors, *Inheritance, Defaults, and the Lexicon*, Studies in Natural Language Processing. Cambridge University Press, pages 38–46.

Evans, R., G. Gazdar, and D. Weir. 1995. Encoding Lexicalized Tree Adjoining Grammars with a Nonmonotonic Inheritance Hierarchy. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 77–84.

Evans, R., G. Gazdar, and D. Weir. 2000. 'Lexical Rules' are just lexical rules. In A. Abeille and O. Rambow, editors, *Tree Adjoining Grammars: linguistic, formal and computational properties*, CSLI Lecture Notes. University of Chicago Press, pages 71–100.

Fellbaum, C. (ed.). 1998. *WordNet an electronic lexical database.* MIT Press.

Gibbon, D. 1992. ILEX: a linguistic approach to computational lexica. In Ursula Klenk, editor, *Computatio Linguae: Aufsatze zur algorithmischen und quantitativen Analyse der Sprache*, volume Beiheft 73 of *Zeitschrift fur Dialektologie und Linguistik*. Franz Steiner, Stuttgart, pages 32–53.

Grimes, B. F. (ed.). 1996. *Ethnologue: Languages of the World.* SIL International, thirteenth edition. also available URL: `http://www.sil.org/ethnologue/ethnologue.html`.

Hansen, P.M. 1990. *Udtaleordbog.* Gyldendal.

Heid, U. and K. Krüger. 1996. A Multilingual Lexicon based on Frame Semantics. In *Proceedings of the AISB-96 Workshop on Multilinguality in the Lexicon*, pages 1–13, Brighton.

Hippisley, A. and G. Gazdar. 2000. Inheritance Hierarchies and Historical Reconstruction: Towards a History of Slavonic Colour Terms. In *Papers of the Chicago Linguistics Society (Main Session)*, pages 125–140, Chicago.

Kameyama, M. 1988. Atomization in grammar sharing. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 194–203.

Kilgarriff, A., L. Cahill, and R. Evans. 1999. The GREG Framework for Multilingual Valency Lexicons. GREG deliverable 2.1, ITRI.

Lock, Z. 2000. Induction and Evaluation of Semantic Networks using Inductive Logic Programming. Master's thesis, School of Cognitive and Computing Sciences, The University of Sussex.

Maas, H.-D. 1987. The MT system SUSY. In M. King, editor, *Machine Translation Today: The State of the Art, Proceedings of the Third Lugano Tutorial, 1984*, pages 209–246, Edinburgh. Edinburgh University Press.

Mahesh, K. 1996. Ontology Development for Machine Translation: Ideology and Methodology. Technical Report MCCS-96-292, Computing Research Laboratory, New Mexico State University.

Menon, B. and N. Modiano. 1993. EAGLES Lexicon Architecture. Technical Report EAG-CLWG-LEXARCH/B, EAGLES.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Nirenburg, S., S. Beale, K. Mahesh, B. Onyshkevych, V. Raskin, E. Viegas, Y. Wilks, and R. Zajac. 1996. Lexicons in the Mikrokosmos project. In *Proceedings of the Artificial Intelligence and Simulated Behavior Workshop on Multilinguality in the Lexicon*, pages 26–33, Brighton, UK. University of Sussex.

Multilingual Glossary of technical and popular medical terms in nine European languages. 1995. Available URL `http://allserv.rug.ac.be/~rvdstich/eugloss/welcome.html`.

Perkins, R.D. 1989. Statistical techniques for determining language sample size. *Studies in Language*, 13(2):293–315.

Pike, K.L. and E.V. Pike. 1947. Immediate Constituents of Mazateco Syllables. *International Journal of American Linguistics*, pages 78–91.

Pinkham, J. 1996. Grammar Sharing between English and French. Technical Report MSR-TR-96-15, Microsoft Research, Redmond.

Rijkhoff, J., D. Bakker, K. Hengeveld, and P. Kahrel. 1993. A Method of Language Sampling. *Studies in Language*, 17(1):196–203.

Ruhlen, M. 1987. *A Guide to the World's Languages; Volume 1: Classification.* Edward Arnold, London.

Ruimy, N., O. Corazzari, E. Gola, A. Spanu, N. Calzolari, and A. Zampolli. 1998. The European LE-PAROLE Project: The Italian Syntactic Lexicon. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 241–248, Granada.

Russell, G., A. Ballim, J. Carroll, and S. Warwick-Armstrong. 1992. A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons. *Computational Linguistics*, 18(2):311–337.

Schubert, K. 1992. Esperanto as an intermediate language for Machine Translation. In J. Newton, editor, *Computers in Translation: A Practical Appraisal.* Routledge.

Slocum, J. 1987. METAL: the LRC machine translation system. In M. King, editor, *Machine Translation Today: The State of the Art, Proceedings of the Third Lugano Tutorial, 1984*, pages 319–350, Edinburgh. Edinburgh University Press.

Slocum, J. and C. Justus. 1985. Transportability to other languages: the natural language processing project in the AI program at MCC. *ACM Transactions on Office Information Systems*, 3(2):204–230.

Smets, M. and R. Evans. 1998. A compact Encoding of a DTG Grammar. In *Proceedings of the 4th Workshop on Tree Adjoining Grammars and Related Formalisms*, pages 164–167, Philadelphia. University of Pennsylvania.

Stump, G.T. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge Studies in Linguistics 93. Cambridge University Press.

Tiberius, C. and L. Cahill. 2000a. Incorporating MetaPhonemes in a Multilingual Lexicon. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 1126–1130, Saarbruecken. Universitaet des Saarlandes.

Tiberius, C and L. Cahill. 2000b. A MetaPhoneme Inventory. In *Computational Linguistics in the Netherlands 1999: Selected Papers from the Tenth CLIN meeting*, pages 193–200, Utrecht, The Netherlands.

Tiberius, C. and R. Evans. 2000. Phonological feature based multilingual lexical description. In *Proceedings of TALN 2000*, pages 347–356, Lausanne, October.

Voegelin, C. 1966. Index to languages of the world. *Anthropological Linguistics*, 8(6-7).

Vossen, P., P. Díez-Orzas, and W. Peters. 1997. Multilingual Design of EuroWordNet. In P. Vossen, N. Calzolari, G. Adriaens, A. Sanfilippo, and Y. Wilks, editors, *Proceedings of the ACL/EACL-97 Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP applications*, Madrid.

Vossen, P. (ed.). 1998. *Special Issue on EuroWordNet*, volume 32. Computers and the Humanities.

Wells, J.C. 1987. Computer-coded phonetic transcription. *Journal of the International Phonetic Association*, 17(2):94–114.

Wells, J.C. 1989. Computer-coded phonemic notation of individual languages of the European Community. *Journal of the International Phonetic Association*, 19(1):31–54.

Wells, J.C. 1995. Computer-coding the IPA: a proposed extension of SAMPA. Available anonymous `ftp:pitch.phon.ucl.ac.uk` in directory `/pub/sam/ipasam-x.ps`, April.

Wolff, S. 1984. The use of Morphosemantic Regularities in the Medical Vocabulary for Automatic Lexical Encoding. *Methods of Information in Medicine*, 23(4):195–203.

Zwicky, A. 1985. How to describe inflection. *BLS*, (11):371–386.

Zwicky, A. 1990. Inflectional morphology as a (sub)component of grammar. In Wolfgang U. Dressler, Hans C. Luschützky, Oskar E. Pfeiffer, and John R. Rennison, editors, *Contemporary Morphology*. Berlin: Mouton de Gruyter, pages 217–236.