

Term Paper

|  |
| --- |
| **The Warrior Stabs the Goat: Creating a Dothraki Resource Grammar with Grammatical Framework** |

**Maria Henkel**

2008709 • maria.henkel@hhu.de

**Benno Kuckuck**

2340409 • benno.kuckuck@hhu.de

October 17th, 2016

|  |  |
| --- | --- |
| **Lecturers:** | Yulia Zinova, Rainer Osswald |
| **Course:** | Introduction to "Grammatical Framework"  SS 2016 |

# Contents

[Contents I](#_Toc463611040)

[List of Figures II](#_Toc463611041)

[List of Tables II](#_Toc463611042)

[1. Introduction 1](#_Toc463611043)

[2. The Project 2](#_Toc463611044)

[2.1 About Resource Grammars and the Standard Library 3](#_Toc463611045)

[2.2 Project Structure and Implemented Categories 4](#_Toc463611046)

[2.3 Operating Instructions and Examples 6](#_Toc463611047)

[3. Discussion 7](#_Toc463611048)

[3.1 Limitations 7](#_Toc463611049)

[3.2 Outlook 7](#_Toc463611050)

[3.3 Conclusion 7](#_Toc463611051)

[Appendix 8](#_Toc463611052)

[Source Code 8](#_Toc463611053)

[References 8](#_Toc463611054)

[Versicherung 10](#_Toc463611055)

## List of Figures

[Figure 1: Mini Resource Grammar Modules and Dependencies (Ranta, 2011a, p. 209) 4](#_Toc463611059)

[Figure 2: GF Resource Grammar Library Categories (Grammatical Framework, 2016b) 6](#_Toc463611060)

## List of Tables

[Table 1: Translation Examples 1](#_Toc462592579)

**

|  |
| --- |
| “Khal Drogo shillo chiories she krazaaj. Chiori fin vijazer adraes zoqwe Khales Drogo. Khal Drogo vinde ma chiories ma adraes she krazaaj. Khal Drogo vachrara.” |
| “Khal Drogo met a woman on a mountain. The woman, who protected a turtle, kissed Khal Drogo. Khal Drogo stabbed the woman and the turtle on the mountain. Khal Drogo stinks.“ |
| GF Run-Time System, 2016 |
|  |

|  |
| --- |
|  |

## Introduction

Grammatical Framework (GF) is a functional programming language, created in 1998 (Ranta, 2011a) and designed to write (natural) language grammars. As GF is “working from a language-independent representation of meaning” (Wikipedia, 2016) it can be used to generate or parse text in different languages at the same time. One of its functions is to translate between languages.

Since GF uses a symbolic approach to process and translate language, in most cases its translations are more precise than those of statistical tools we use in everyday life (Brown & Frederking, 1995). For example, the “Google Translate” service (Google, 2016) will produce ungrammatical sentences as soon as the input is too complicated while the GF translator (Grammatical Framework, 2016a) has no problems producing a correct translation (see Table 1) in many cases.

Table 1: Translation Examples

|  |  |  |
| --- | --- | --- |
| Input | Output Google Translate | Output GF Translator |
| I heal these women. | Ich diese Frauen zu heilen. | |  |  | | --- | --- | |  | Ich verheile diese Frauen. | |
| The warrior will kiss that cheese behind the tree. | Der Krieger wird Kuss, der Käse hinter dem Baum. | Der Krieger wird jenen Käse hinter dem Baum küssen. |
| Diese heiße Frau heilt diese Königin im Meer. | This hot woman heals the queen of the sea. | This hot woman heals this queen in the sea. |

This also means, that GF – other than Google Translate – depends on a large knowledge base of predefined grammar rules to achieve high-quality language processing: Its own standard library, the GF Resource Grammar Library, covers more than 38 different languages (Ranta, 2015) - although many of them are not yet complete - and is still growing. In 2010, it already consisted of more than 500,000 lines of code (Ranta, 2011a).

As Grammatical Framework is an open-source project (Ranta, 2009) and still under development, everyone is invited to use it or even help further develop it. For example, by adding languages to the library, which is the aim of the project described in this paper.

Our goal is to write a GF resource grammar for the Dothraki language and, eventually, be able to add it to the GF Resource Grammar Library. The work described in this paper is meant to lay a foundation for this goal (see Chapter 2).

Dothraki is a constructed language, developed by the language creator and writer David J. Peterson (Peterson, 2016a). The Dothraki are a fictional race of nomadic horse warriors in the “A Song of Ice and Fire” book series by George R. R. Martin (Martin, 2016). As the series was adapted for television by the television network HBO, the language bits found in Martin’s books had to be extended to a fully functional language (Peterson, 2016b): The Dothraki language. Today, it has a vocabulary of over 3,000 words (Tongues of Ice and Fire Wiki, 2016). We will show a few examples of the Dothraki language grammar in this paper, but more information can be found in the book “Living Language Dothraki” (Peterson, 2014), on Peterson’s blog (2016b) and on the Tongues of Ice and Fire Wiki (2016) among many other sources.

This paper aims to serve as a short documentation and explanation of our project. It will cover the project aim and scope, general information regarding the project as well as its structure and explanations of sample functions and the corresponding Dothraki grammar rules (Chapter 2). We will also discuss problems and limitations and briefly describe future plans before we summarize and evaluate the project in its current state and our experience while working on it (Chapter 3).

## The Project

As mentioned before, the aim of this project is to create a resource grammar for the Dothraki language and add it to the GF Resource Grammar Library, so that other Dothraki fans and/or GF enthusiasts may use it. While there are currently a few Dothraki language learning apps and online translators avaialable (e.g. Cognitus Apps, 2016; Fun Translations, 2016), they are still very limited and lack the quality and range to actually learn the language or provide an accurate translation. By the means of an open-source Dothraki resource grammar, better tools like these or even different projects could be created more easily.

### About Resource Grammars and the Standard Library

The GF Resource Grammar Library covers the morphology and basic syntax of several languages. Here, the idea is to predefine "program details, which require expert knowledge, and make them available for non-expert application programmers" (Ranta, 2009). This way the library users need not have the "specialized linguistic expertise" (Ranta, 2009) to write an application grammar in GF. Considerable work has been and is being done by the GF developers and resource grammar programmers to make this possible:

*The idea of using a grammar as a software library is, to our knowledge,*

*new in GF. It has required a considerable effort in the design and implementation of the GF programming language: a type system, a module system, compilation techniques, and optimizations. [...] The effort*

*made in this work is supplemented by a substantial effort in the library itself.*

*The code included in the library is more than twice in size compared to the*

*implementation of GF.* (Ranta, 2009)

The provision of the GF Resource Grammar Library as an easy-to-use tool may, on the one hand, be more time consuming, but, on the other hand, enables more people to use it - even without the expert knowledge. Ranta (2011, p. 200) also recommends to add new languages to the library, instead of directly writing an application grammar for them, as they can “serve an unlimited number of applications” and because “the existing library specification will help to identify the linguistic issues and avoid pitfalls.” He estimates, that an interval of three to six months of full-time work is needed for an experienced GF programmer to implement a new resource grammar (p. 222).

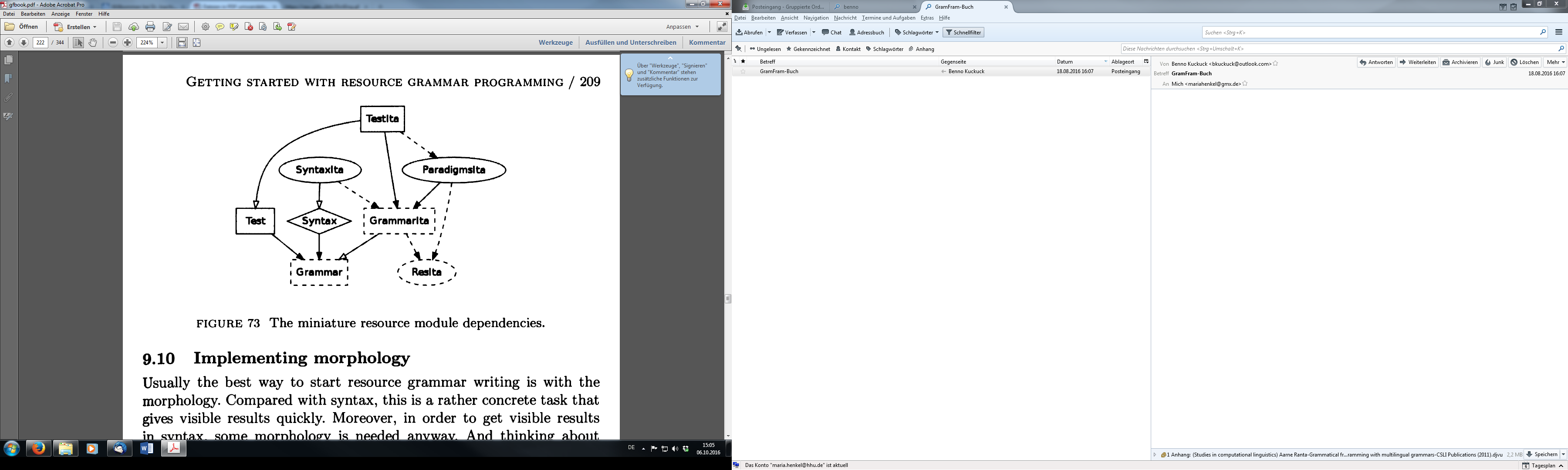


Figure 1: Mini Resource Grammar Modules and Dependencies (Ranta, 2011a, p. 209)

As the workload of building a full resource grammar exceeds the time frame for this term paper, the first big milestone of our project was to create a mini resource grammar, similar to the Italian mini resource grammar, an example of Ranta's book (2011, pp. 237-245, see Figure 1). This first milestone is to lay the foundation for a full resource grammar. Other than in Ranta's example, we already use the categories and structure of a full resource grammar (see Figure 2) to make a later expansion easier.

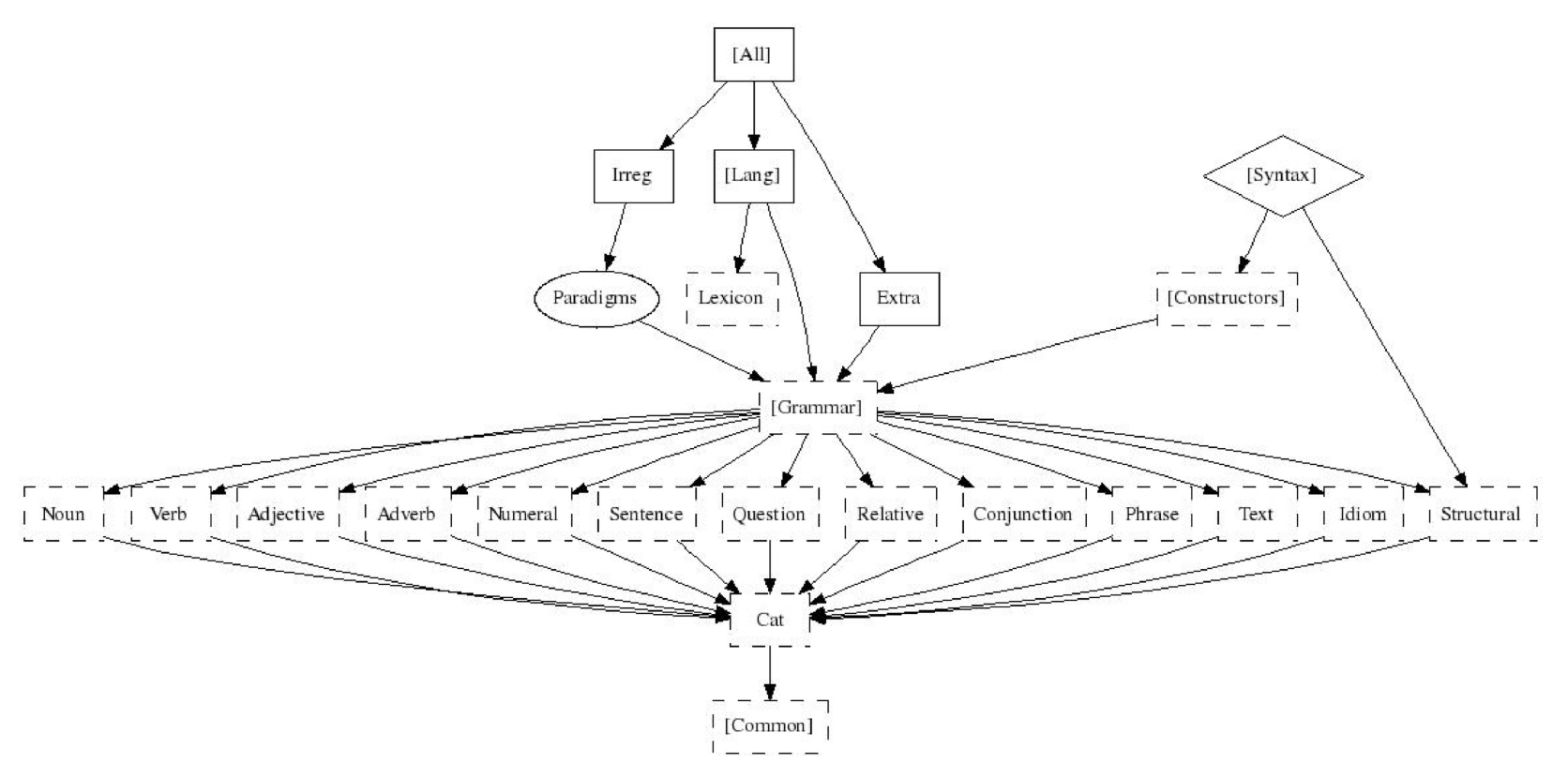


Figure 2: Module Structure of a Full Resource Grammar (Ranta, 2011b)

### Project Structure and Implemented Categories

Just like the GF repositories (Détrez & Camilleri, 2016), our full project is hosted on GitHub (Henkel & Kuckuck, 2016) so that others may see, use and contribute to it. It consists of the following folders:

* simpleapi
* simpleabstract
* simpledothraki
* simpleenglish
* common
* example
* documentation
* vocabulary

The “simpleapi” folder (like the “api” folder in the full resource grammar library) contains the outward facing part of the library – the types and functions available to application grammar developers. Those are implemented in terms of more basic functions, which are declared in the abstract grammar(s) contained in the “simpleabstract” folder. For each language, in our case just English and Dothraki, there is one folder (here: “simpledothraki” and “simpleenglish”) which contains the concrete implementations of these abstract grammars. Some of these parts, those shared by all languages, are implemented in the “common” folder. We are planning to remove the “simple” from all folder- and filenames as soon as the resource grammar is complete. Since we already use the structure of a full resource grammar, it should work accordingly. It should be noted, that we are working with the “simpleenglish” resource grammar, to be able to translate Dothraki phrases into English for testing and presenting purposes. Later on, the full English resource grammar, already available via the GF Resource Grammar Library, will be used.

The mini resource grammar can already be used for application grammars. The “example” folder contains application grammar rules to show how the resource grammar can be applied.

There are two more folders: The “documentation” folder is holding two versions of our documentation, one markdown file for readers on GitHub and one print version. Finally, the “vocabulary” folder contains the vocabulary extractor, a supporting python script, written by us to add a relatively large number of words to our dictionary automatically. It takes the source code of the Dothraki vocabulary site (Ice and Fire Wiki, 2016) as input, extracts words and their translations from it and transforms them into the right format, needed for the resource library. For example, the entry "ador [aˈdor], ni. chair" on the website will create the three output strings: "ador\_N = mkN "ador" inanimate;", "chair\_N = mkN "chair";" and "ador\_N = chair\_N;" – one entry for the Dothraki linearization, one entry for the English linearization and one entry for the dictionary. This way, we were able to add more than 1644 words to our dictionary, and will be able to add even more as soon as the resource grammar is expanding.

Figure 3 shows the main categories of the GF Resource Grammar Library and the current status of our implementation. The yellow categories have been fully or at least partially implemented in our resource grammar. For the first milestone, we do not aim to implement every single instance of every category. Instead, we want to first implement the most important categories in a proper way, to make later expansion easier. For example, there are 32 different instances of the “mkCl” function for the clause category “Cl” (Grammatical Framework, 2016b). We did not implement all of them in the current version of our resource grammar, but since there are already some “mkCl” functions included, it is easier to add more functions of the same type later on.

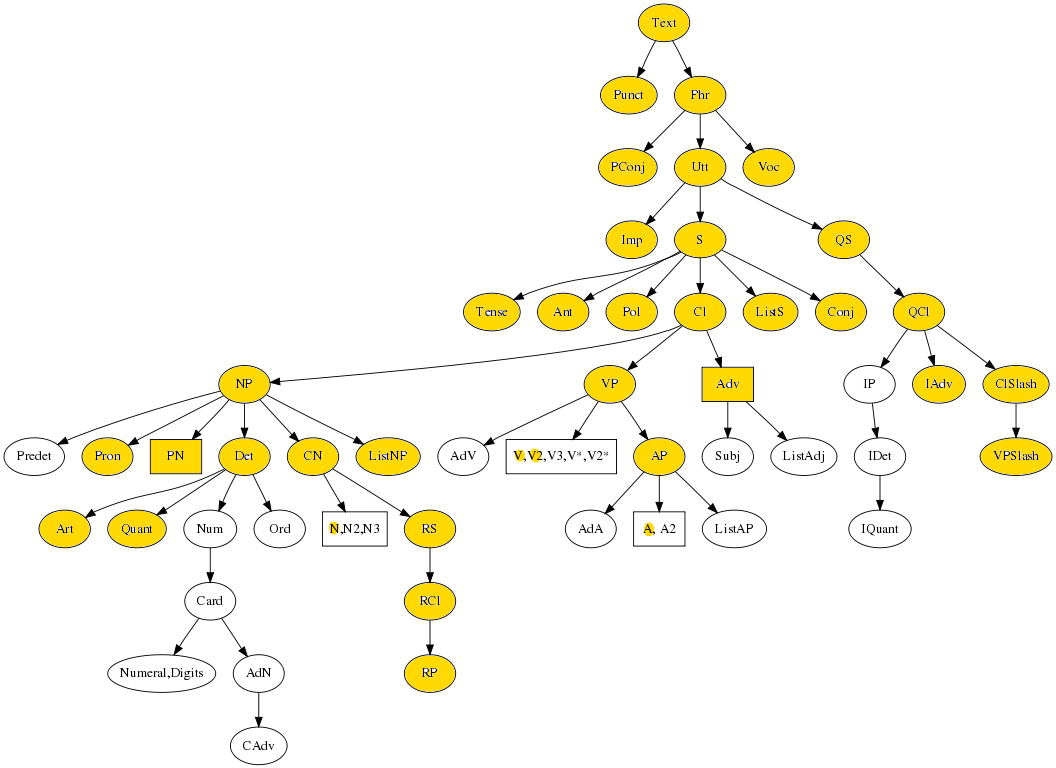


Figure 3: GF Resource Grammar Library Categories (Grammatical Framework, 2016b)

With the current implementation status, it is already possible to parse and translate a wide range of phrases. The following example text is fully supported by the current version of the Dothraki resource grammar:

Beispieltext…

### Operating Instructions and Examples

Wie benutzen : application grammar und/oder lang

CatSimpleDot + Dothraki Language

Beispielstücke erklären? Besonderheiten der Sprache?

Sprachliche Besonderheiten: Verben 🡪 Polarität, etc.

Beispiele der Ausgabe.

## Discussion

### Limitations

Since .. language… a complicated and lengthy task,

Dothraki: Nicht alles vollständig dokumentiert/existent, deshalb auch nicht möglich zu implementieren. (Folie 198, good grammar book needed) zB Strukturwörter

Not linguists, learning process, beginners

Warum nicht alles auf einmal? Zu viel Arbeit, alles gleichzeitig, keine Möglichkeit zu testen, deshalb Mini resource, Ziel: später zur richtigen aufbauen.

Regeln versuchen so viel wie mögl. Abzudecken, aber nicht immer möglich 🡪 siehe Buch? Zitat?

Grenze erreicht von sachen die dokumentiert sind.

### Outlook

Expand grammar to a full resource grammar. Community?

### Conclusion

Sinnvolles projekt, ambitioniertes ziel

## Appendix

### Source Code

The complete source code can be viewed and downloaded via GitHub:

<https://github.com/mahen2/dothraki_gf>

## References

Brown, R., & Frederking, R. (1995). Applying statistical English language modeling to symbolic machine translation. In *Proceedings of the sixth international conference on theoretical and methodological issues in machine translation (TMI-95)* (pp. 221-239).

Cognitus Apps (2016). *Flamingo Dothraki.* Retrieved from https://play.google.com/store/apps/details?id=com.cognitusapps.flamingo.dothraki&hl=de

Fun Translations (2016). *Dothraki translator.* Retrieved from <http://funtranslations.com/dothraki>

Détrez, G. & Camilleri, J. J. (2016). *Grammatical Framework*. GitHub.com. Retrieved from https://github.com/GrammaticalFramework

Google (2016). *Google Translate*. Retrieved from <https://translate.google.com/>

Grammatical Framework (2016a). Translation with GF: Powered by multilingual grammars. Retrieved from <http://www.grammaticalframework.org/demos/translation.html>

Grammatical Framework (2016b). GF Resource Grammar Library: Synopsis. Retrieved from http://www.grammaticalframework.org/lib/doc/synopsis.html

Martin, G. R. R. (2016). Bibliography. Retrieved from <http://www.georgerrmartin.com/bibliography/>

Henkel, M. & Kuckuck, B. (2016). *Building a resource grammar for Dothraki*. GitHub.com. Retrieved from https://github.com/mahen2/dothraki\_gf

Peterson, D. J. (2014). *Living Language Dothraki: A Conversational Language Course Based on the Hit Original HBO Series Game of Thrones*. New York: Living Language.

Peterson, D. J. (2016a). *The Art of Language Invention*. Retrieved from <http://www.artoflanguageinvention.com/>

Peterson, D. J. (2016b). About Dothraki. Retrieved from <http://www.dothraki.com/about-dothraki/>

Ranta, A. (2009). The GF Resource Grammar Library. *Linguistic Issues in Language Technology*, 2(2), pp. 1-63.

Ranta, A. (2011a). *Grammatical Framework: Programming with Multilingual Grammars.* Stanford: CSLI Publications.

Ranta, A. (2011b). Grammatical Framework: Programming with Multilingual Grammars. Slides for the GF book. Retrieved from http://www.grammaticalframework.org/gf-book/gf-book-slides.pdf

Ranta, A. (2015). *The Status of the GF Resource Grammar Library.* Grammatical Framework. Retrieved from <http://www.grammaticalframework.org/lib/doc/status.html>

Tongues of Ice and Fire Wiki (2016). Vocabulary. Retrieved from http://wiki.dothraki.org/Vocabulary

Wikipedia (2016). Grammatical Framework. Retrieved from https://en.wikipedia.org/wiki/Grammatical\_Framework

## Versicherung

Hiermit versichern wir, diese Arbeit selbständig verfasst und keine anderen als

die angegebenen Quellen benutzt zu haben. Wir versichern ferner, dass diese

Arbeit oder Teile davon nicht schon für eine andere Abschlussprüfung oder einen anderen Beteiligungsnachweis eingereicht wurden. Uns ist bewusst, dass ein Zuwiderhandeln gegen diese Versicherung eine Ordnungswidrigkeit darstellt, die mit einer Geldbuße geahndet werden kann.

Düsseldorf, den 17.10.2016

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_