Institut für Sprache und Information
Heinrich-Heine-Universität Düsseldorf
Computational Morphology
Dr. Julia Zinova
Hausarbeit
SoSe 2020

Modelling the morphology of Wymysorys

LUKAS MIELCZAREK

lukas.mielczarek@uni-duesseldorf.de
2756129
Computerlinguistik BA, 4. Fachsemester
September 22, 2020

Contents

Diagrams

1 About the language

Wymysorys is a West-Germanic language spoken in the small town Wilamowice in the south of Poland (near Silesia). It is also known as Vilamovicean and called Wymysiöeryś by its speakers.

Wymysorys is sometimes called the most endangered Germanic language in the world. Practically only old people speak it – except for one person, the co-author of a grammar of Wymysorys Tymoteusz Król, who learned Wymysorys form an old lady and is trying to revitalize it teaching it to young people.

Only up to 20 people are proficient in Wymysorys with almost all native speakers born before 1930.

Although many scolars see Wymysorys as a dialect or a colonial variety of German, others – especially recently – argue that it should be regarded as a proper language. Recently Wymysorys got its own language code and a Language Academy for Wymysorys was established in Warsaw.

The inhabitants of Wilamowice emphasize their own linguistic and cultural uniqueness and dissimilarity from German. They claim that in contrast to other colonial German varieties spoken in the area were always regarded as German dialects while Wymysorys never was.

Moreover the language is not mutually understandable with Standard German and has its own unique features in phonology, lexicon and morphology – some of which I came across during my project.

1.1 History

Although the Wymysorys participated in the High German Sound Shift, researchers claim that it exhibits Felmish, Anglo-Saxon and/or Frisian traits. Many Vilamovians think that their ancestors came from Flanders, Holland or Friesland. Some even claim to be of Anglo-Saxon descent.

Certain traits of the language are said to distinguish Wymysorys from East Central German and to support the claim that Wymysorys is of West German descent. The palatalization of

velar stops, loss of the constant *n* in unstressed syllables and the h-form of the third person masculine singular pronoun *har* seem to support that (see Andrason, Król, 2016).

One possible theory in regards to the descendence of the language speculates that a big group of settlers may have come from the area between Main and Rhine and a smaller group from the Netherlands, Flanders and Wallonia in 12th century.

In an area with such a rich linguistic history as South Poland, it is obvious that Wymysorys also had a lot of contact with the Polish language throughout the centuries. Even before the establishment of the modern Polish Republic and during Austrian and German occupation a lot of Polish loanwords found their way into the language. The close contact also boosted the development of some features very unique to a Germanic language.

Especially in recent times, due to the decline in native speakers and the opression of the language in communist times (from WW2 up to the 90s), the number of Polish loanwords has sky-rocketed.

Currently there are several revitalization projects in place to preserve the language and bring it back to life.

1.2 Written tradition

Written works in Wymysorys are a fairly recent phenomenon. The frist written work appeared only in 1860 (SOURCE) and used a very different orthography than the one in place today. Currently a lot of original and translated work is being made available. It follows the modern othography developed by (SOURCE). I personally own a copy of „The little Prince“ in Wymysorys.

I will base my work around the orthography described in THE GRAMMAR. My goal is to use a few sentence from „The little Prince“ as input for my morphological analysis.

1.3 Material

I used several sources for information about the language.

Grammar of Wymysorys: The most extensive Grammar of Wymysorys. Based on hundreds of hours of recordings containing translations, conversations, stories, songs, poems, prayers and more. The authors Alexander Andrason and Tymoteusz Król documented phonetic, morphological and syntactical phenomena. LINK

1.4 Morphology

Wymysorys exhibits a typical indo-european morphology. It is synthetic. Affixes of different categories fuse together – like NUM and CASE.

Wymysorys morphology is not far away from German but has a lot of specific differences.

1.4.1 Articles

Articles function the same as in German. They can be inflected in gender, number and case.

They also appear in clitic form e. g. as suffix for prepositions. Such compounds are can be observed are far more often than in Standard.

See the followin example for Reference. The symbol ^ marks the morpheme boundary.

Standard German		Wymysorys		English
seperate	compound	seperate	compound	
bei dem	bei^m	by ym	bym	at the (masculine)
bei der	non existent	by yr	byr	at the (feminine)

Table 1: Clitic Articles in Wymysorys

1.4.2 Nouns

Nouns in Wymysorys can appear in several case forms just like in German. These cases are nominative, accusative, dative, genitive and vocative. Unlike German the genetive is only used with person names and certain other words. The vocative – not present in Standard German – is not productive and only found in a hand full of words.

In regards to gender, masculine, feminine and neuter nouns are found in Wymysorys just like in German. Nouns can appear in singular and plural forms.

There are three classes of declension for each gender. In most cases the word recieves some case specific affix in the plural but none in the singular forms. So they are often underspecified. Accusative plural and nominative plural tend to be the same while dative plural gets a special ending.

Many words exhibit an important morphological change when building the plural besides affixation: Umlaut/Ablaut. When changing into plural a vowel of the standard (singular) form of the word changes into an other vowel. This usually happens in a specific pattern – e.g. *a* regularlily changes to *e*. But there are a lot of irregularities.

1.4.3 Adjectives

Adjectives appear in six different classes depending on the context the adjective is used in. Every adjective can be declined in each of the six classes. They are all to some degree productive.

e.g.

(1) a. nist gütys
nothing good-CLASS1
nothing good
nichts Gutes

b. à güt màkja
INDEF good-CLASS5 girl
a good girl
ein gutes Mädchen

As you can see the form of the adjective depends on the context. In Standard German the two cases are phonetically identical.

There are two types of adjective declension: Adjectives ending in *a* and the standard class. Adjectives of the first type do not show any ending except in one of the six context classes. Adjectives of the standard class recieve endings in all of the forms (context class, gender, number, case) – although many of the affixes are GLEICHLAUTEND ZUSAMMENFALLEND RICHTIGES WORD shared morphemes. In the plural form adjectives do not show any difference between the three genders.

2 Modelling the language

Adjectives, nouns, articles and prepositions.

2.1 Strategy

My approach is based on a classical lexc-strategy. The file `Grammar.txt` contains the Lexicon and classes __ rules ___.

The script `Main.fst` is responsible for importing the lexc-file and composing the net. The necessary rules for phonological changes happening at the morpheme boundary, override exceptions and additional, unproductive cases are split into seperate scripts found in the Rules-folder (`PhonologicalRules.fst`, `OverrideExceptions.fst`, `AddtionalCases.fst`) for maintanance reasons.

The python script `testsuit.py` can be used to automatically test the script against language data and as a general tool for working with the Wymysorys xfst project.
See chapter 4 for more information about the python script.

2.2 Exceptions

Currently there are two override exceptions found in the script `OverrideExceptions.fst`. They replace two ill-formed words which would otherwise be generated by the script.

Analysis	Ill-formed	Correct
kynd+Noun+N+sg+gen	kynds	kyndyś
kyng+Noun+M+pl+dat	kynga	kyngja

Table 2: Override Exceptions

I tried to minimize such cases and to model most of the observations into productive rules. Since I did not have an extensive morphological description of the language or phonotactics available, not every single interpretation I performed might be completely correct.

Therefore I created the test suit (see chapter 4) very early on during my project – to continuously check if necessary adjustments in the phonological rules preserve the correct description of the rest of the linguistic facts.

2.3 Articles

I used the multichar symbol *+Article* to denote articles. Articles are split into two categories: definite *+def* and indefinite *+indef*. I split them into two seperate lexicons in the Grammar.txt-file: **Art_def** and **Art_indef**.

Grammar.txt

```
> LEXICON Articles
> der+Article+def:0        Art_def ;
> der+Article+indef:ȧ      Art_indef ;
```

der+Article+def is mapped to the empty word since the definite article forms do not share a common beginning while the indefinite forms a built using the base *ȧ* and adding a suffix.

Der is used as the lexeme for the adjectives.

2.3.1 Definite

The singular *+sg* definite forms are all seperately listed under the lexicon **Art_def** while the plural *+pl* are built using the lexicons **Art_def_pl1** and **Art_def_pl2** since they have no distinction for gender. I still marked them for gender since the word type as a whole exhibits this category.

```
> LEXICON Art_def
> +M+sg+nom:der            # ;
> +M+sg+acc:yn             # ;
…
> +N+sg+gen:yr             # ;
…
> +F+sg+dat:yr             # ;
>                          Art_def_pl1 ;
```

The three genders masculine, neuter and feminine are marked using by *+M*, *+N*, *+F*. They are marked before the number to follow the same convention I used with nouns. There the gender is marked before the number since gender is a category inherent to the specific noun.

The number category is marked before the case (*+nom, +acc, +gen, +dat*), since Wymysorys words that decline for number and case often exhibit different case behaviour in the plural. This way it is possible to seperate the plural into a seperate lexicon just like in this case.

```
> LEXICON Art_def_pl1
> +M:0     Art_def_pl2 ;
> +F:0     Art_def_pl2 ;
> +N:0     Art_def_pl2 ;
```

Since definitice articels do not have dinstinctive forms for different genders in the plural, I matched the gender markers all with the empty word in **Art_def_pl1** and set the continuation class for all three genders to **Art_def_pl2**.

```
> LEXICON Art_def_pl2
> +pl+nom:dy               # ;
…
> +pl+gen:der              # ;
> +pl+gen:yr               # ;
```

Similar to the singular forms the seperate forms for the cases are listed here. As you may notice the genitive can be expressed by *der* or by *yr*. Therefore +pl+gen: matches with both forms.

2.3.2 Indefinite

The gender- and case-specific suffixes get appended onto *ȧ* in the lexicon **Art_indef**. There are only singular indefinite articles. There were no genitive forms listed in the sources I used.

2.4 Prepositions

As you may expect prepositions do not inflect in any category. They are still pretty interesting since they can be suffixed by some definite articles in accusative and dative form.

Prepositions can be found in the **Prepositions** lexicon. They all refer to the continuation class **Prep_suff**. When standing alone without an article suffix they just receive the *+Preposition* marker. When used with an article suffix they get marked by *+Preposition+Cl+Article+def* with *+Cl* indicating the suffixation by a clitic article.

The definitive articles are phonologically reduced in their clitic form (e.g. feminine dative *-r* instead of *yr*), exhibit additional forms only found as a clitic (e.g. masculine accusative *-a*) and certain forms are missing as a clitic variant (e.g. neuter accusative *-s* exists, while feminine accusative cannot be constructed). Due to these facts I modelled the clitics as distinct morphemes in their own lexicons instead of using the regular definite article lexicon as a continuation class and reducing the articles phonologically later on using the `PhonologicalRules.fst` script.

```
> LEXICON Prep_suff
> +Preposition+Cl+Article+def:0   Art_suff_def ;
…
> LEXICON Art_suff_def
>               Art_suff_def_sg ;
>               Art_suff_def_pl1 ;
>
> LEXICON Art_suff_def_sg
> +M+sg+acc:^a    # ;
…
>
> LEXICON Art_suff_def_pl1
> +M:0    Art_suff_def_pl2 ;
…
>
> LEXICON Art_suff_def_pl2
> +pl+dat:^a      # ;
> +pl+dat:^n      # ;
```

The structure is analogous to the lexicons used for the regular adjectives with the plural forms not distinguishing gender.

The morpheme boundary between the preposition and the clitic adjective is marked with ^. This is necessary since several prepositions experience phonological changes when being suffixed.

1. Irregular *yn*-forms

Observation:

a) *yn* ‘in’ + *m* ‘the’ [masculine/neuter sg dative] → *ym* instead of **ynm* or **ynum* (analogous to observation 2)

b) *yn* ‘in’ + *a/n* ‘the’ [masculine sg accusative | pl dative] → *yn* instead of **yna* (analogous to observation 4)

c) *yn* ‘in’ + *r* ‘the’ [feminine sg dative] → *yr* instead of **yner (*analogous to observation 3)

d) *yn* ‘in’ + *s* ‘the’ [neuter sg accusative] → *ys* instead of **yns*

Analysys:

When receiving a suffix *yn* drops the *n* and behaves as if *y* was the stem.

Consequence:

Introduction of a multicharacter symbol *a* marking the ending of a word dropped during suffixation. Therefore the dictionary entry now reads *y/an*. *n* is marked as the ending of the word to drop. Now I had to add a regular expression in the script `PhonologicalRules.fst` to remove the ending when some character follows the morpheme boundary ^.

In cases without a suffix, the multicharacter symbol */a* is removed by the rule Elim.

`PhonologicalRules.fst`

```
...
> define Phon1 [ %/a n %^ -> 0 || _ ? ] ;
…
> define Phon Phon1    …     ;
…
> define Elim [ {^} -> 0 , %/a -> 0 , %/oe -> 0 ] ;
…
> define PhonologicalRules  ...  Phon .o. Elim ;
```

In order for the upper side of *y/an* to be *yn* I had to add two filters to the `Main.fst` script:

Main.fst

```
...
> source < Rules/PhonologicalRules.fst
>
> regex PhonologicalRules ;
>
> read lexc < Grammar.txt ;
>
>
> # Filter
>
> regex [ 0 -> %/a || ? _ ? ] ;
>
> regex [ ~[?*] <- $[ %/a ] ] ;
>
> #
>
> compose net ;
…
```

regex [0 -> %/a || ? _ ?] ; allows insertion of */a* into any spot within a word when converting from upper to lower side before consulting the lexicon rules while regex [~[?*] <- $[%/a]] ; maps upper strings containing */a* to the null language, which deletes them.

2. insertion of -*u*-

Observation:

a) *uf* ‘on’ + *m* ‘the’ [masculine/neuter sg dative] → *uf^m* → *ufum*

b) *myt* ‘with’ + *m* the’ [masculine/neuter sg dative] → *myt^m* → *mytum*

Analysys:

When a morpheme ending in *f* or *t* gets followed by a morpheme starting with *m* the symbol *u* is inserted between the two.

Consequence:

I added the rule `define Phon2 [ %^ -> u || f _ r , t _ r ] ;` to the script to replace the morpheme boundary symbol ^ with *u* in such cases.

3. insertion of -*e*-

Observation:

a) *uf* ‘on’ + *r* ‘the’ [feminine sg dative] → *uf^r* → *ufer*

b) *myt* ‘with’ + *r* ‘the’ [feminine sg dative] → *myt^r* → *myter*

Analysys:

When a morpheme ending in *f* or *t* gets followed by a morpheme starting with *r* the symbol *e is inserted between the two.*

Consequence:

I added the rule `define Phon3 [ %^ -> e || f _ r , t _ r ] ;` to the script to replace the morpheme boundary symbol ^ with *e* in such cases.

4. Application of -*n*/-*a*-form

Observation:

a) *uf* ‘on’ + *a/n* ‘the’ [masculine sg accusative | pl dative] → *ufa*

b) *by* ‘on’ + *a/n* ‘the’ [masculine sg accusative | pl dative] → *byn*

Analysys:

The -*a* suffix-variant [masculine sg accusative | pl dative] is only used with prepositions ending in a consonant while those that end in a vowel get suffixed by the -*n* suffix-variant in such cases.

Consequence:

I added a filter that is composed into the network after the step performing the phonological rules. The filter ensures that only strings not containing an *n* following a consonant and a morpheme boundary or an *a* following a vowel and a morpheme boundary are well-formed.

PhonologicalRules.fst

```
...
> # Phonological Rules
> define Post1 ~[ $[ [ C - l - ł ] %^ n ] | $[ V %^ a ]
      ] ;
...
> define Post Post1 .o. Post2 ;
…
> define Phon Phon1 .o. Phon2 .o. Phon3 .o. Phon4 .o.
      Phon5 .o. Phon6 .o. Phon7 .o. Phon8 .o. Phon9 .o.
      Phon10 .o. Phon11 .o. Phon12 .o. Post ;
…
```

In this case *l* and *ł* are excluded from the consonant set since *l^n* and *ł^n* are a valid combinations as we will see later.

The consontant set *C* and the vowel set *V* are defined at the beginning of the file.

```
…
> define V [ e | i | ȧ | a | o | u | ü | y | ö ] ;
> define C [ p | b | t | d | k | j | k | g | f | w | s
      | z | ś | ź | ż | h | n | ń | m | r | l | ł |
      ć ] ;
```

5. Certain forms cannot be constructed in the neuter singular accusative

a) *uf* ‘on’ + *s* ‘the’ [neuter sg accusative] → *ufs*

b) *fjy* ‘before’ + *s* ‘the’ [neuter sg accusative] → *fjys*

c) by ‘at’ + *s* ‘the’ [neuter sg accusative] → non existant

Analysys:

I could not find any regularity in this behaviour. Therefore I concluded that these forms are unproductive and can only be found in a hand-full of cases.

Consequence:

I removed the neuter singular accusative form from the lexicon and added the four forms *ys, ufs, fjys a*nd *nös* found in THE GRAMMAR to the script file /Rules/AdditionalCases.fst. They are later unfied with the general network.

```
AdditionalCases.fst
...
> define Prepositions [ {yn}:{ys} | {uf}:{ufs} | {fjy}:
     {fjys} | {nö}:{nös}] %+Preposition:0 %+Cl:0 %
     +Article:0 %+def:0 %+N:0 %+sg:0 %+acc:0 ;
...
```

6. Unexpected behaviour of *undyr*

Observation:

a) *undyr* ‘under’ + *m* ‘the’ [masculine/neuter sg dative] → *underum* | *undum* instead of **undyrum* (analogous to oberservation 2)

b) *undyr* ‘under’ + *r* ‘the’ [feminine sg dative] → *undyr* | *underer* instead of **undyrer* (analogous to oberservation 3)

c) *undyr* ‘under’ + *a/n* ‘the’ [masculine sg accusative | pl dative] → *undyn* instead of **undyra* (analogous to observation 4)

Analysys:

c) When *r* is followed by a morpheme boundary and *n* or *a*, it gets replaced with the empty word. In this case this creates *undy^n* and **undy^a*. According to obversation 4 only the former is well-formed. Later findings seem to support such an analysis.

b) When a morpheme ending with *yr* gets followed by a morpheme starting with *r* then *yr* needs to be replaced by *y* or *ere*. I do not have any scientific evidence for such an analysis although I still have not encountered any indication for it not being applicable for certain cases. Therefore I used it in my network.

a) When a morpheme ending with *yr* gets followed by a morpheme starting with *m* then *yr* needs to be replaced by *ere* or *u*. Just as before, I do not have any scientific evidence for such an analysis although I still have not encountered any indication for it not being applicable for certain cases. Therefore I used it in my network.

Consequence:

I added these specific rules into the `PhonologicalRules.fst`-file:

PhonologicalRules.fst

```
...
> define Phon4 [ r -> 0 || _ {^n} , [ ? - e ] _ {^a} ,
_ {^s} ] ;
> define Phon5 [ {yr^} -> [ {y} | {ere} ] || _ r ] ;
> define Phon6 [ {yr^} -> [ {eru} | {u} ] || _ m ] ;
...
```

Later findings confirmed rule **Phon4** also being useful when r gets followed by a morpheme boundary and s. Furthermore I observed an exception to the rule: r remains when followed by a only if it gets preceded by an e (er-suffix, see REFERENCE).

The current rules work for all cases know to me.

2.5 Nouns

Wymysorys nouns belong to one of three declension types (1. -Ø, 2. -a, 3. -n). They determine the way in which a noun is inflected in the categories number and case (i.e. the set of endings they receive).

Since the occurrence of the gender markers (*+M, +F, +N*) depends on the noun, I had to assign 9 different basic continuation classes to the nouns in my **Nouns** lexicon:

Masculine: **N_M1, N_M2, N_M3**

Feminine: **N_ F1, N_F2, N_F3**

Neuter: **N_N1, N_N2, N_N3**

I later had to split the lexicon **N_M2** into two: **N_M2_c1** and **N_M2_c2**. More information on that later.

The basic classes lead to lexicons matching the *+Noun* marker and the appropriate gender marker and then continuing with the according general lexicon for the declension types: **N_1, N_2, N_3**.

```
Grammar.txt
...
> LEXICON Nouns
> !masculine
> kyz           N_M1 ;
…
> LEXICON N_M1
> +Noun+M:0     N_1 ;
…
> LEXICON N_F1
> +Noun+F:0    N_1 ;
...
> LEXICON N_1
...
```

Besides gender nouns also inflect for number and case. The set of case markers differs depending on the noun being in singular or in plural form. Therefore I marked number (*+sg, +pl*) first and then case (*+nom, +acc, +dat, +gen*). Since nouns in the plural only show one morpheme – either a plural morpheme (with the case morphemes all being empty) or a case morpheme (with the plural morpheme being empty) depending on the inflection class – and since there also is no visible singular morpheme, one could reject my analysis arguing that number and case are expressed in one common morpheme. I still hold my analysis to be appropriate for reasons of redundance reduction. For cases where the plural morpheme is visible (e. g. **N_2_pl** +pl:^a/d, see the following diagram) my analysis allows to declare it only once in the grammar. Analysing it as part of a number-case morpheme would mean that you had to mention it several times in the grammar (+pl+nom:^a/d, +pl+acc:^a/d, …).

The following diagram shows the whole noun system (to be read from left to right). There are a lot of syncretisms and identical forms in the three declension classes which I analysed as common lexicons to ease maintanance of the program. They are marked using the same cell colour.

The lexicons are written in **bold** while the matching strings are written in *cursive* script.

<table>
<tr><td rowspan="4">N_M1
+Noun+M:0</td><td rowspan="4">N_1</td><td rowspan="4">N_sg</td><td rowspan="4">+sg:0</td><td rowspan="4">N_case_sg</td><td>+dat:0</td><td colspan="3">#</td></tr>
<tr><td rowspan="3">N_nom_acc_sg</td><td>+acc:0</td><td>#</td><td colspan="2"></td></tr>
<tr><td colspan="2" rowspan="2">N_nom_sg</td><td>+nom:0</td><td>#</td></tr>
<tr><td>+gen:^s@R.GE</td><td>#</td></tr>
</table>

							N.ALLOWED@
N_F1 *+Noun+F:0*		*+pl:/d*	**N_sp_case_pl**	*+dat:^A*	#		
N_N1 *+Noun+N:0*				**N_nom_acc_pl**	*+nom:0*	#	
					+acc:0	#	
N_M2_c1 *+Noun+M:0*	**N_M2_c1_num**	*+sg:0*	**N_M2_c1_sg**	**N_M2_dat_sg**	*+dat:^a*	#	
				N_nom_acc_sg	see above		
		N_2_pl		*+pl:^a/d*	**N_case_pl**	*+dat:0*	#
						N_nom_acc_pl	see above
N_M2_c2 *+Noun+M:0*	**N_M2_c2_num**	*+sg:0*	**N_M2_c2_sg**	**N_M2_dat_acc_sg**	*+acc:^a*	#	
					N_M2_dat_sg	see above	
				N_nom_sg	see above		
		N_2_pl		see above			
N_F2 *+Noun+F:0*	**N_2**	**N_sg**		see above			
		N_2_pl		see above			
N_N2 *+Noun+N:0*							
N_M3 *+Noun+M:0*	**N_3**	**N_sg**		see above			
N_F3 *+Noun+F:0*		*+pl:^n/d*	**N_case_pl**	see above			

N_N3 *+Noun* *+N:0*				

Table 3: Overview of the Noun System

2.5.1 Plural

Nouns in the inflectional class 1 derive their plural with no ending. Class 2 nouns have a common *a*-suffix in the plural while nouns of class three share the ending *n* for all plural forms.

Therefore for the Lexicon **N_1** the plural marker *+pl* only matches with the multicharacter symbol */d* (for the reasons behind this, see Ablaut) while in **N_2** the marker matches with with *^a/d* (*a*-Suffix). In **N_3** the plural marker matches with *^n/d*. The ^ marks the morpheme boundary.

Since the special lexicons for the two types of declension class 2 masculine nouns **N_M2_c1** and **N_M2_c2** differ in case markings for forms in singular number, they do not forward to **N_2** but to their own lexicons responsible for the number selection **N_M2_c1_num** and **N_M2_c2_num.** Since they share the same plural-case-pattern as the regular **N_2** nouns in feminine or neuter, I created a seperate lexicon for the plural marking of class 2 nouns that **N_M2_c1_num, N_M2_c2_num** and **N_2** forward to and where the plural-marker is matched with the suffix to reducide redundancy: **N_2_pl.**

```
Grammar.txt
...
> LEXICON N_2
>                N_sg ;
>                N_2_pl ;
>
> LEXICON N_M2_c1_num
> +sg:0          N_M2_c1_sg ;
>                N_2_pl ;
>
> LEXICON N_M2_c2_num
> +sg:0          N_M2_c2_sg ;
>                N_2_pl ;
…
> LEXICON N_2_pl
> +pl:^a/d       N_case_pl ;
...
```

Irregularities:

Class 1

There were no irregularities observable with class 1 nouns.

Class 2

ja-Plural:

Various nouns of class 2 do not express the plural using the *a*-ending but rather the form *ja* (*ryk-ja* ‘backs’ instead of **ryk-ja*, *tȧjhja* ‘ponds’ instead of **tȧjha*, *Pulkja* ‘Poles’ instead of **Pulka*). This always occurred in the GRAMMAR when the stem was ending in the consonants *k* or *h*. Therefore I added a phonological rule that replaces the morpheme boundary ^ with a *j* when preced by a *k* or an *h* and followed by an *a*.

```
PhonologicalRules.fst      - Version 1 -
...
> define Phon7 [ %^ -> j || [ k | h ] _ a ] ;
...
```

Furthermore there are several other words, ending in a vowel, which experience *ja*-plural: e.g. *kni* ‘knee’ → *knija* instead of **knia*, *he* ‘hay’ → *heja* instead of *hea*. Therefore I had to modify the rule to apply to stems ending in a vowel:

```
PhonologicalRules.fst      - Version 2 -
...
> define Phon7 [ %^ -> j || [ k | h | V ] _ a ] ;
...
```

Then, I had to exclude *y* from the vowel set since suffixed prepositions ending in *y* were ill-formed into invalid forms like *yn* ‘in’ → **yja* (masculine singular accusative) or *fjy* ‘before’ → **fjyja*. As explained in the section about prepositions, those variants should be filtered out.

```
PhonologicalRules.fst      - Version 3 -
...
> define Phon7 [ %^ -> j || [ k | h | V - y ] _ a ] ;
...
```

Exception from ja-Plural with semivowels:

The plural form of the word *zöu* ‘pig’ is not **zoüja* as the phonological rule **Phon7** might suggest – placing a *j* in every morpheme boundary where a vowel and an *a*-suffix meet. But the correct plural form is *zoüa*. I think this is the case since the *ü* acts as a semivowel in the *oü*-diphtong.

Therefore I included a list of descending diphthongs, **DescDiph**, that end in semivowels and modified the **Phon7**-rule and the **Post1**-rule which filters the vowel-^*a* combination:

```
PhonologicalRules.fst
...
> define DescDiph [ a i | e i | o ü | o i ] ;
…
> define Post1 ~[ $[ [ C - l - ł ] %^ n ] | $[ V %^ a ]
                          ] | $[ DescDiph %^ a ] ;
...
> define Phon7a [ %^ -> 0 || DescDiph  _ a ] ;
> define Phon7b [ %^ -> j || [ k | h | V - y ] _ a ] ;
…
> define Phon Phon1 .o. Phon2 .o. Phon3 .o. Phon4 .o.
Phon5 .o. Phon6 .o. Phon7a .o. Phon7b .o. Phon8 .o.
Phon9 .o. Phon10 .o. Phon11 .o. Phon12 .o. Post ;
```

The **Post1** expression still filters all vowel-boundary-*a* combinations but does include every string containing a descending vowel followed by a morpheme boundary and an *a* due to the addition of | $[DescDiph %^ a].

I split the **Phon7** rule into two: **Phon7a** and **Phon7b**. The one matches a morpheme boundary with the empty word when it is preceded by a descending diphtong and followed by an *a* or an *e*. Therefore such combinations are not object to the second rule which simply contains the old **Phon7**-expression since they do not contain a morpheme boundary anymore.

This solution works flawlessly with all the cases currently covered by the test suit.

Class 3

łn/ln-Plural:

Since several nouns of all genders ending in *l* or *ł* exhibit the *n*-Suffix for plural, I had to modify the phonological rule I introduced with the Prepositions that prohibits the occurrence of a consonant followed by a morpheme boundary and an *n*:

```
PhonologicalRules.fst      - Version 1 -
...
> # Phonological Rules
> define Post1 ~[ $[ C %^ n ] | $[ V %^ a ] ] ;
...
```

This first version did not recognized words like *śtejłn* ‘handles’ or *engln* ‘angels’ due to the *łn/ln*-combination. Therefore I removed *l* and *ł* from the set:

```
PhonologicalRules.fst      - Version 2 -
...
> # Phonological Rules
> define Post1 ~[ $[ C %^ n ] | $[ V %^ a ] ] ;
...
```

guł → gln-irregularity:

The word *föguł* ‘bird’ has an unusual behaviour when put into plural form. The plural form is not **fögułn* but *fygln*. The vowel change from *ö* to *y* is an observation that I describe further down as **Ablaut**. The change from *guł* to *gln* required an additional rule that performs this change in case the *guł* is followed by a morpheme boundary and an *n*.

```
PhonologicalRules.fst      - Version 1 -
...
> define Phon10 [ {gułˆ} -> {gl} || _ n ] ;
...
```

Annihilation of r:

There is a subgroup of words in class 3 that have a stem ending in *r*. In the plural form this *r* is not visible: e.g. *rür* ‘pipe’ → *rün* instead of **rürn*, *popjyr* ‘paper’ → *popjyn* instead of **popjyrn*.

Luckily, the removal of the *r* is already described by rule **Phon4**.

```
PhonologicalRules.fst      - Version 2 -
...
> define Phon4 [ r -> 0 || _ {^n} , [ ? - e ] _ {^a} ,
_ {^s} ] ;
...
```

The morpheme boundary is preserved to help process the phenomenon described in the next section.

yn-Plural:

There are certain words ending in *-er* that experience a substitution of this *-er*-ending with *y* when getting the *-n*-Plural-suffix (e.g. *tełer* ‘plate’ → *tełyn* instead of **tełern*, *foter* ‘father’ → *fatyn* instead of **fotern*, *müter* ‘mother’ → mütyn instead of **mütern*).

One could analyse this change as a substitution of *er^n* with *yn* but it seems to be more promising to build on the *r*-annihilation-rule established in the last step. This rule already tells us to process a word like *tełer* in the following way: *tełer^n → tełe^n*.

Now, only a rule substituting *e^* with *y* when followed by an *n* is needed:

```
PhonologicalRules.fst
...
> define Phon8 [ e -> y || _ % ^ [ n | s ] ;
...
```

Later findings suggest that this rule is also applicable when the *e* is followed by ^s.

Now the script perfectly tackles the irregularities while at the same time not messing with the established processes. The process now goes like this: *tełer^n → tełe^n → tełyn*.

Gap-y-Plural:

There are several words of class 3 whose stem does not end in *-er* but that receive the ending *-yn* in the plural: *hün* ‘hen’ → *hinyn, śłājs* ‘torch’ → *śłājsyn, śłap* ‘slipper’ → *śłapyn*.

As you can see, all of these words have a stem ending in a consonant. Therefore one could analyse the *y* as a filler vowel that fills the morpheme boundary when a consonant and an *n* meet.

Therefore I introduced a phonological rule responsible for performing this change:

PhonologicalRules.fst

```
...
> define Phon11 [ %^ -> y || [ C - l - ł - f ] _ n] ;
...
```

As we have seen earlies, *l* and *ł* need to be excluded form the consonant set since plurals in *ln* and *łn* are possible.

Furthermore, I had to exclude *f* from the set – due to the fact that the prepositions *uf* ‘ on’ when suffixed with *n*, does not convert in the way *uf^n* → *ufyn*. No *y*-insertion takes place when *f* precedes *n*, but rather the *fn*-consonant combination is filtered out by the **Post1** filter rule described in the section about prepositions.

PhonologicalRules.fst

```
...
> define Phon11 [ %^ -> y || [ C - l - ł - f ] _ n] ;
...
```

Suppletive Roots

Some nouns derive their plural using suppletive roots: e.g. *fojermon* ‘fireman’ → *fojerłoüt* ‘firemen’, *cymermon* ‘carpenter’ → cymerłoüt ‘carpenters’. I would have added two seperate lexicon entries for every such word (including singular and plural markers) but the GRAMMAR did not include information about the appropriate inflection classes for the singular and the plural roots of these words. Further reasearch did not help me either. Therefore I did not know which case-selection-lexicons the roots would need to continue with. So I did not include them in my grammar.

2.5.2 Ablaut

In all of the three declension classes there is a subtype of words which experience a change in a vowel of the root when in plural form. This phenomenon is called Ablaut or Umlaut. It occurs in all declension types but only with masculine and feminine words (exception: declension type 2 masculine).

The change of vowel (or diphtong) follows certain regular patterns. E.g. *u* often changes to *y*. But for certain vowels there are several possible alternations entirely depending on the root word: *o* in *kroft* (strength) changes to *e* while it changes to *a* in *foter* (father).

Furthermore, there are words where the Ablaut-feature is simply optional (e.g. sg *fan* 'banner' - pl *fan* or *fen*).

To model this vowel change I introduced several multicharacter symbols marking vowels within nouns in the lexicon to be replace by phonological rules:

/u, /2u, /3u

(*u* standing for Umlaut)

They each stand for different replacement sets. Furthermore I added a parallel set of multicharacter symbols marking voews within nouns in the lexicon to be optionally replaces by phonological rules:

/ou, /2ou, /3ou

(ou standing for optional Umlaut*)*

Lastly I declared the multicharacter symbol */d* (short for 'do'). It gets matched with the plural marker *+pl* in the lexicons **N_1, N_2_pl** and **N_3**.

Dictionary entries for words experiencing the Ablaut feature had to be modified from *con* to *c/3uon* and from *śejn* to *ś/uejn*.

Then I added a set of regular expressions in the script `PhonologicalRules.fst` to convert vowels marked with one of the ablaut-markers with the appropriate replacement vowel if the marker */d* can be encountered somewhere after the ablaut-marker in the word.

PhonologicalRules.fst

```
...
> define Ablaut1 [ %/ou -> [ %/u | 0 ] ] ;
> define Ablaut2 [ u -> y , a -> e , o -> e , {iöe} ->
          ȧ , ö -> y , ü -> i , {ej} -> y || %/u _ ?*
          %/d ] ;
> define Ablaut3 [ %/2ou -> [ %/2u | 0 ] ] ;
> define Ablaut4 [ o -> a , ü -> {jy} , e -> {ȧj} ,
          {iöe} -> a || %/2u _ ?* %/d ] ;
> define Ablaut5 [ %/3ou -> [ %/3u | 0 ] ] ;
> define Ablaut6 [ o -> yy || %/3u _ ?* %/d ] ;
> define Ablaut7 [ %/u -> 0 , %/d -> 0 , %/2u -> 0 , %/
          3u -> 0 ] ;
> define Ablaut Ablaut1 .o. Ablaut2 .o. Ablaut3 .o.
          Ablaut4 .o. Ablaut5 .o. Ablaut6 .o.
          Ablaut7 ;
…
> define PhonologicalRules Ablaut .o. Dat .o. Phon .o.
          Elim ;
```

The optional markers are matched with normal markers or the empty word.

The three replacement categories are not based on any scientific measure but only on the order I made observations about such replacement patterns. E.g. I first read that *o* in *kroft* (strength) changes to *e* and included it in the first replacement pattern. Then I encountererd the change of *o* to *a* in *foter* (father) and had to establish the second replacement class */2u.*

The rule **AblautElim** deals with deleting the markers after the ablaut-alternations took place. This is necessary in order for other phonological modifications (e.g. dealing with morpheme boundaries) to work correctly. Therefore the Ablaut-rules are the first rules in the order of the phonological modifications.

In order for the upper sides of the words modified with the ablaut-markers in the lexicons to show without them I had to add filters to the `Main.fst` script (analogous to prepositions obervation 1):

Main.fst

```
...
> source < Rules/PhonologicalRules.fst
>
> regex PhonologicalRules ;
>
> read lexc < Grammar.txt ;
>
>
> # Filter
>
…
> regex [ 0 -> %/u || ?* _ ? ] ;
>
> regex [ 0 -> %/2u || ?* _ ? ] ;
...
> regex [ 0 -> %/ou || ?* _ ? ] ;
...
> regex [ ~[?*] <- $[ %/u ] ] ;
>
> regex [ ~[?*] <- $[ %/2u ] ] ;
...
>
> #
>
> compose net ;
…
```

In total there are six rules of type `regex [ 0 -> %/u || ?* _ ? ] ;` and six rules of type `regex [ ~[?*] <- $[ %/u ] ] ;`. The first ones allow insertion of */u* (*/2u, /3u* …) into any spot within a word (including at the front; necessary due to the word */2ue – e* 'egg') when converting from upper to lower side before consulting the lexicon rules while the second ones map upper strings containing these markers to the null language, which deletes them.

2.5.3 Case

Of the five Wymysorys cases – nominative, accusative, dative, genitive and vocative – only three – nominative, accusative and dative – are productive. For these cases I used the same straight-forward continuation-lexicon approach I used for the plural, as you can seen in the diagram above.

Case, Singular:

Case, Singular, regular pattern:

For most nouns, mainly all nouns belonging to class 1 and 3 and all feminine and neuter nouns of class 2, the case-endings are marked in the same way. I used the continuation lexicon **N_sg** for this purpose. **N_1, N_2** and **N_3** all forward to this class:

<table>
<tr><td rowspan="4">N_sg</td><td rowspan="4">+sg:0</td><td rowspan="4">N_case_sg</td><td>+dat:0</td><td>#</td><td colspan="3"></td></tr>
<tr><td colspan="2" rowspan="3">N_nom_acc_sg</td><td>+acc:0</td><td colspan="3">#</td></tr>
<tr><td rowspan="2">N_nom_sg</td><td>+nom:0</td><td colspan="2">#</td></tr>
<tr><td>+gen:^s+gen:^s@R.GEN.ALLOWED@</td><td colspan="2">#</td></tr>
</table>

Table 4: Singular Noun Case Declension

First, in **N_sg** the singular marker gets matched with an empty morpheme. Then, there is a three-step process involving three different lexicons: **N_case_sg** (from which the regular dative form is expressible and which also forwards to **N_nom_acc_sg**), **N_nom_acc_sg** (from which the regular accusative form is expressible and which also forwards to **N_nom_sg**) and **N_nom_sg** (from which the regular nominative and genitive forms are expressible.

I implemented the case system in this manner since the two variants of class 2 masculine nouns share subsets of the regular case endings. Therefore creating seperate lexicons expressing these subsets reduces redundancy.

Case, Singular, irregular masculine class 2 pattern:

Masculine nouns of class 2 experience a different behaviour for singular case endings than their feminine and neuter class 2 counterparts. Therefore it was necessary to distinguish them form the **N_2** class and create distinct classes for the two subtypes: **N_M2_c2_num** and **N_M2_c1_num**.

The irregularity for singular case endings is already apparent by the fact that they are marked. In the regular pattern they are not expressed. Here, the first subtype expresses a dative ending *a* while maintaing the rest of the cases unexpressed (underspecified?). The second subtype not only expresses dative with an *a*, but also accusative.

Therefore I established the **N_M2_dat_acc_sg** lexicon, forwarded to by **N_M2_c2_sg**. There the first option is to match the accusative marker *+acc* with *^a*. The second option forwards to **N_M2_dat_s** which is shared with the first subclass **N_M2_c1_sg**, matching the dative marker *+dat* with *^a*.

The first subclass **N_M2_c1_sg** also forwards to **N_nom_acc_sg** where it takes the regular path for the nominative, accusative and genitive cases. While **N_M2_c2_sg** forwards to **N_nom_sg** where it takes the regular path only for nominative and gentive since the accusative is also expressed in an irregular way as described.

<table>
<tr><td rowspan="5">N_M2_c1_num</td><td rowspan="3">+sg:0</td><td rowspan="3">N_M2_c1_sg</td><td colspan="2">N_M2_dat_sg</td><td>+dat:^a</td><td>#</td></tr>
<tr><td colspan="3">N_nom_acc_sg</td><td colspan="2">see above</td></tr>
<tr><td colspan="2" rowspan="2">N_2_pl</td><td rowspan="2">+pl:^a/d</td><td rowspan="2">N_case_pl</td><td>+dat:0</td><td>#</td></tr>
<tr><td>N_nom_acc_pl</td><td>see above</td></tr>
<tr><td rowspan="5">N_M2_c2_num</td><td rowspan="4">+sg:0</td><td rowspan="4">N_M2_c2_sg</td><td colspan="2" rowspan="2">N_M2_dat_acc_sg</td><td>+acc:^a</td><td>#</td></tr>
<tr><td>N_M2_dat_sg</td><td>see above</td></tr>
<tr><td colspan="3">N_nom_sg</td><td>see above</td></tr>
<tr><td colspan="2">N_2_pl</td><td colspan="3">see above</td></tr>
</table>

Table 5: Irregular Masculine Singular Noun Case Declension

ja-form:

When the stem ends with *k* or *h* and gets followed by the *a*-case-marker, a *j* is inserted into the morpheme boundary. This process also occurs when the plural marker is expressed by *a* as previously described. Therefore I have already established the means to deal with this change thanks to rule **Phon7**.

The fact that this change also occurs when *a* is an entirely different morpheme (with the same phonetic realization), further strengthens the analysis that this irregularity is of phonetic nature and that there is no seperate *ja*-morpheme.

Case, Singular, Genitive:

The genitive case has significantly lost productivity. It is limited to certain nouns like kinship terms. But it is still productive for names. Therefore I included the genitive in my analysis and introduced the Flag Diacritics *@U.GEN.ALLOWED@* and *@R.GEN.ALLOWED@* to deal with it. I declared the two Flag Diacritics as multicharacter symbols in the Grammar.txt file and marked all nouns that can appear in genitive case with *@U.GEN.ALLOWED@*. E.g.:

Grammar.txt

```
...
> LEXICON Nouns
...
> @U.GEN.ALLOWED@śiłer     N_M3 ;
> @U.GEN.ALLOWED@f/2uoter  N_M3 ;
...
> @U.GEN.ALLOWED@müter     N_F3 ;
...
> @U.GEN.ALLOWED@kynd      N_N3 ;
...
> @U.GEN.ALLOWED@jüza      N_M1 ;
> @U.GEN.ALLOWED@tüma      N_M1 ;
...
```

The genitive is marked with *s* for all classes. Therefore I established a second option in the lexicon **N_num_sg** (reached by the singular nouns of all classes):

N_nom_sg	*+nom:0*	#
	+gen:^s@R.GEN.ALLOWED@	#

Table 6: Noun Genitive Singular

The Flag Diacritic @R.GEN.ALLOWED@ ensures that the GEN-feature is set to ALLOWED – which is only the case for nouns that are declared to allow the genetive.

Lastly I needed to include the command set obey-flags ON ; in the Main.fst script.

yś-Genitive:

Words ending in *er* substitute their last two letters with *yś* to declare the genitive instead of appending an *s* – e.g. *foter* ‘father’ → *fotyś* instead of **foters*, *müter* ‘mother’ → *mütyś* instead of **müters*.

This phenomenon strongly resembles the *yn*-plural form of words ending in *er* in inflection class 3. Therefore I made use of the existing rules for the handling of the *er*→*yn*-plural: **Phon4** and **Phon8**. When I introduced them, I already stated that it makes sense to make use of them in cases where there is not an *n* following the morpheme boundary but rather an *s:*

PhonologicalRules.fst

```
...
> define Phon4 [ r -> 0 || _ {^n} , [ ? - e ] _ {^a} ,
           _ {^s} ] ;
…
> define Phon8 [ e -> y || _ %^ [ n | s ] ;
...
```

Now, first the *r* is eliminated and then the substitution from *e* to *y* performed:

foter^s → *fote^s* → *foty^s*

But as you see, *fotys* does still not resemble the goal *fotyś* 100%. Therefore I included an additional rule, matching *y^s* with *yś*:

PhonologicalRules.fst

```
...
> define Phon9 [ {y^s} -> {yś} ] ;
...
```

Now the procedure functions as follows:

foter^s → *fote^s* → *foty^s* → *fotyś*

Until now, this strategy has not led to ill-formed words and wrong analyses or changes in the subset of cases covered by my grammar so far.

Irregular case: kynd

Despite not ending in *er*, the word *kynd* ‘child’ has the genitive form *kyndyś* instead of **kynds*. This suggests a similar analysis as with nouns of class 3 ending in a consonant: the

insertion of *y* as a gap filler. Then, the same process as above converts the *y^s* to *y^ś*. The whole process should read:

kynd^s → kyndy^s → kyndyś

Conversion 2 is already established by **Phon9**. Conversion 1 can build on the existing rule **Phon11**:

```
PhonologicalRules.fst      - Version 1 -
...
> define Phon11 [ %^ -> y || [ C - l - ł - f ] _ n] ;
...
```

I had to include the *s* as a possible context:

```
PhonologicalRules.fst      - Version 2 -
...
> define Phon11 [ %^ -> y || [ C - l - ł - f ] _ [ n |
          s ] ] ;
...
```

Now conversion 1was set into place. *kynd^s* got converted to *kyndys*. But this modification was still not enough. In order for rule **Phon9** to detect the pattern, there needed to be a morpheme boundary between *y* and *s*. So I modified the rule once more:

```
PhonologicalRules.fst      - Version 3 -
...
> define Phon11 [ %^ -> y %^ || [ C - l - ł - f ] _ [ n
          | s ] ] ;
...
```

Now, one last modification was needed: **Phon11** needs to take place before **Phon9** (i.e. before *y^s* is matched with *yś*. Therefore I swapped the order of the order of the two rules. Due to consistency I held on to the names. I will need to enact a proper naming scheme in a future revision.

```
PhonologicalRules.fst      - Version 4 -
...
> define Phon11 [ %^ -> y %^ || [ C - l - ł - f ] _ [ n
          | s ] ] ;
…
> define Phon9 [ {y^s} -> {yś} ] ;
…
> define Phon Phon1 .o. Phon2 .o. Phon3 .o. Phon4 .o.
          Phon5 .o. Phon6 .o. Phon7a .o. Phon7b .o.
          Phon8 .o. Phon11 .o. Phon10 .o. Phon9 .o.
          Phon12 .o. Post ;
```

Now, *kynd+Noun+N+sg+gen* got matched with *kyndyś* perfectly fine.

Case, Plural:

Case is rarely marked in the plural. Nominative and Accusative are unmarked for all of the inflection classes. Therefore the lexicon **N_nom_acc_pl** is reachable by **N_1, N_2, N_3** and also **N_M2_c1_num** and **N_M2_c2_num**.

N_nom_acc_pl	*+nom:0*	#
	+acc:0	#

Table 7: Noun Plural Case Markers

There are no genetive forms in the plural. Genitive expressions are instead marked analytically with the help of prepositional constructions. Therefore I did not have to include them in the lexicon.

The only case distinguishable in the plural, is the dative.

Case, Plural, Dative:

For plural nouns of class 1, dative is marked by the morpheme *-a*. But there is an exception to the rule: The ending is not present for nouns whose stem ends in *-n*, *-a* and *-um*. Therefore I had to add a phonological rule that would match the *-a* suffix with the empty word in an environment where *a* is preceded by *n*, *a* or *um* and a morpheme boundary.

This procedure, however, was not successful since this also meant that the -a suffix would get deleted in cases where the a would represent a different morpheme (e.g. plural suffix). Therefore I had to mark the -a-dative suffix in a different way for the phonological rule to

detect only these dative cases. I sticked with an A in upper letters. The script converts it to an empty word if it is preceded by n, a or um and a morpheme boundary or to an a in all other cases:

```
PhonologicalRules.fst
...
> # Dative A
> define Dat1 [ {^A} -> 0 || n _ , a _ , {um} _ ] ;
> define Dat2 [ {^A} -> {^a} ] ;
> define Dat Dat1 .o. Dat2 ;
...
> define PhonologicalRules Ablaut .o. Dat .o. Phon .o.
      Elim ;
```

The lexicon N_sp_case_pl matches this functional ^A-suffix with the dative marker +dat. It represents the continuation lexicon for N_1 when it matches the plural marker:

N_1	**N_sg**		**...**		
	+pl:/d	**N_sp_case_pl**	*+dat:^A*	#	
			N_nom_acc_pl	*+nom:0*	#
				+acc:0	#

Table 8: Noun Plural Dative Special Marker

The other classes **N_2, N_3, N_M2_c1_num** and **N_M2_c2_num** get forwarded to a different lexicon **N_case_pl** when matched with the plural marker. There option 1 is matching the dative marker with an empty word and option 2, forwarding to the already established **N_nom_acc_pl** lexion, where the nominative and accusative cases can be realised. This forwarding is again done to reduce redundancy.

N_case_pl	*+dat:0*	#
	N_nom_acc_pl	see above

Table 9: Noun Plural Dative Regular Marker

kyngja:

The plural dative of the word *kyng* ‘king’ is *kyngja* instead of **kynga*. However, this does not seem to be a hint at a possible application of a regular rule inserting a *j* into the morpheme boundary – for example **Phon7b** – since other words ending in *g* that get appended by an *a*-suffix do not experience this change: *oüg* ‘eye’ → *oüga* and not **oügja*.

Therefore I included *kyngja* as an exceptional analysis that overwrites the ill-formed analysis for *kyng+Noun+M+pl+dat*:

```
OverrideExceptions.fst
...
> define OverrideExceptions [ {kyng}:{kyngja} %+Noun:0
%+M:0 %+pl:0 %+dat:0 ] ;
```

Case, Vocative

The vocative case is not productive in Wymysorys – neither in the singular, nor in the plural. Only a hand full of words have vocative forms. Therefore I manually added them in the Rules/AdditionalCases.fst file:

```
Rules/AdditionalCases.fst
...
> # extra definitions for words in vocative
>
> define VocativeSG [ müm %+Noun:0 %+F:0 | büw %+Noun:0
      %+M:0 | bow %+Noun:0 %+N:0 | pot %+Noun:0 %+N:0 |
      knȧht      %+Noun:0 %+M:0 ] %+sg:0 %+voc:y ;

> define VocativePL [ łoüt %+Noun:0 %+M:0 ] %+pl:0 %
      +voc:y ;
…
> define AdditionalCases VocativeSG | VocativePL |
      Prepositions ;
```

As you can see, the only vocative form I could find is expressed by a *y*.

2.6 Adjectives

Adjectives can inflect in case, number and gender by appending suffixes – just like nouns. Therefore I used the same markers for these categories in the same order as before (1.

gender: *+M, +F, +N*, 2. number: *+sg, +pl*, 3. case: *+nom, +acc, +dat*). The genitive case does not occur.

There are six different declensional classes for adjectives. The class determines the set of suffixes for the three categories case, number and gender. The difference in comparison to nouns is that the class is not an inherent category of the adjective. Every adjective can be inflected in each of the six classss. The choice of class is determined by the context the adjective is used in (e.g. predicative use, used together with a definite article etc).

Therefore I first introduced six class markers (*+c1, +c2, +c3, +c4, +c5, +c6*). Later I had to add the markers *+c5a* and *+c5b* due to further nuances in the declension pattern the GRAMMAR pointed out.

These markers have their place before the three other categories since they determine the specific version of an ending they match with. Therefore an adjective analysis might look like this: *kośer+Adjective+c3+F+pl+dat*.

The categories gender, number and case and fusional. For each combination of them, there is one specific suffix that gives information about all three categories.

Furthermore, adjectives can be inflected for degree. For simplicity of notation, I decided that the positive form shall not be marked in any special way while the comparative recieves the marker *+comp*. For more information about my decisions regarding the comparative, see the section _____.

In general all adjectives decline in the same way. There is only a small subset of adjectives with a distinctive declension pattern and there are some irregular adjectives with suppletive roots that I will also cover later. But the great majority of adjectives all have the same declension pattern. Therefore I could introduce one common continuation class for nearly all of the adjectives in my lexicon: **adj_marker**. There the adjective marker *+Adjective* gets matched with the empty word and the class gets continued with **adj_degree**. There, for positive adjectives, as already stated, no ending gets appended and the continuation lexicon **adj_classes** gets into play, where the class selection takes place.

Grammar.txt

```
...
> bywájt      adj_marker ;
> güt           adj_marker ;
> śejn          adj_marker ;
…
> LEXICON adj_marker
> +Adjective:0            adj_degree ;
>
> LEXICON adj_degree
>                         adj_classes ;
> +comp:/comp/d       adj_comp ;
>
> LEXICON adj_classes
> +c1:0    adj_c1 ;
> +c2:0    adj_c2 ;
> +c3:0    adj_c3 ;
> +c4:0    adj_c4 ;
> +c5:0    adj_c5 ;
> +c5a:0   adj_c5a ;
> +c5b:0   adj_c5b ;
> +c6:0    adj_c6 ;
```

The 6 (8) classes share a lot of common morphemes and in a way form a continuum only modifiying a few forms going from class to class – from c1 (strong declension) to c6 (weak declension). Therefore I put a lot of forms into common lexicons reached by more than inflection class. In a way this is an assumption of underspecification. This also makes maintanance easier: If I need to alter the ending *-a* due to some morpho-phonological reasons, I just need to look at one spot.

The following diagram shows the whole adjective system (to be read from left to right). Common lexicons are only declared once and then referred to using the same cell colour. The lexicons are written in **bold** while the matching strings are written in *cursive* script.

The names of the lexicons were meant to be systematic, but certainly need some modification to be easier understandable.

<table>
<tr><td rowspan="4">adj_marker

+Adjective:/radj

adj_</td><td rowspan="4">adj_classes</td><td rowspan="4">+c1:0</td><td rowspan="4">adj_c1</td><td rowspan="4">adj_c1_sg</td><td rowspan="4">adj_c1_c4_sg</td><td rowspan="4">adj_c1_c2_c3_c4_sg</td><td colspan="2">+M+sg+nom:^er</td><td>#</td></tr>
<tr><td colspan="2">+M+sg+acc:^a</td><td>#</td></tr>
<tr><td rowspan="2">adj_c1_F</td><td>+F+sg+nom:^y</td><td>#</td></tr>
<tr><td>+F+sg</td><td>#</td></tr>
</table>

degree										+acc:^y	
						+N+sg+nom:^ys				#	
						+N+sg+acc:^ys				#	
					adj_c1_sg_dat_com	+M+sg+dat:^um	#				
						+F+sg+dat:^er	#				
						+N+sg+dat:^um	#				
				adj_c1_pl	+M:0	**adj_c1_pl_marker**	+pl:0	**adj_c1_pl_case**	+nom:^y	#	
					+N:0				+acc:^y	#	
					+F:0				+dat:^a	#	
		+c2:0	**adj_c2**	**adj_c1_sg**	see above						
				adj_c2_N	+N+sg+nom:0	#					
					+N+sg+acc:0	#					
				adj_c1_pl	see above						
		+c3:0	**adj_c3**	**adj_c3_c5_sg**	**adj_c2_N**	see above					
					adj_c1_c2_c3_c4_sg	see above					
				adj_c1_pl	see above						
				adj_c1_sg_dat_com	see above						
		+c4:0	**adj_c4**	**adj_c1_c4_sg**	see above						
				+M:0	**adj_c4_sg_dat_MN**				+sg+d	#	

				+N:0			*at:^am*	
				adj_c4_sg_dat_com	*+M:0*	**adj_c4_sg_dat_com2**	*+sg+dat:^a*	#
					+N:0			
					+F:0			
				adj_c1_pl	see above			
		+c5:0	**adj_c5**	**adj_c1_pl**	see above			
				adj_c4_sg_dat_com	see above			
				adj_c3_c5_sg	see above			
				+F+sg+nom:/oe	#			
				+F+sg+acc:/oe	#			
				+F+sg+dat:/oe	#			
		+c5a:0	**adj_c5a**	*+F+sg+dat:^y*	#			
			adj_c5	see above				
		+c5b:0	**adj_c5b**	*+M+sg+nom:^y*	#			
			adj_c5a	see above				
		+c6:0	**adj_c6**	**adj_c4_sg_dat_com**	see above			
				adj_c1_F	see above			
				+M+pl:0	**adj_c4_pl2**	*+nom:^a*	#	
				+F+pl:0		*+acc:^a*	#	
				+N+pl:0				
				adj_c1_pl	see above			
				+M+sg+nom:^y	#			
				+M+sg+acc:^a	#			
				+N+sg+nom:^y	#			
				+N+sg+acc:^y	#			

<table>
<tr><td rowspan="2"></td><td></td><td></td><td></td><td></td><td>+F+sg+dat:^y</td><td>#</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="2">+comp:/d</td><td colspan="3">adj_comp</td><td colspan="5">See table 2 (under development)</td></tr>
<tr><td rowspan="8">adj_marker

+Adjective:/aadj</td><td rowspan="8">adj_classes_a</td><td>+c1:0</td><td rowspan="7">adj_gen_a</td><td rowspan="3">+M:0</td><td rowspan="7">adj_num_a</td><td>+pl:0</td><td rowspan="7">adj_case_a</td><td>+nom:0</td><td>#</td></tr>
<tr><td>+c2:0</td><td rowspan="2">+gen:0</td><td rowspan="2">#</td></tr>
<tr><td>+c3:0</td><td rowspan="2">+N:0</td></tr>
<tr><td>+c5:0</td><td>+sg:0</td></tr>
<tr><td>+c5a:0</td><td rowspan="3">+F:0</td><td>+dat:0</td><td>#</td></tr>
<tr><td>+c5b:0</td></tr>
<tr><td>+c6:0</td></tr>
<tr><td>+c4:0</td><td>adj_c4_a</td><td>adj_gen_a</td><td colspan="5">see above</td></tr>
<tr><td></td><td></td><td>+M+sg+acc:^n</td><td colspan="5">#</td></tr>
<tr><td></td><td></td><td>+M+pl+dat:^n</td><td colspan="5">#</td></tr>
</table>

Table 10: The Adjectival System

The six classes have different degrees of frequency in use. Some are nearly unproductive whereas others are very common. Since you can still almost always inflect adjectives in all of the classes, I decided to inlcude all of them in my grammar.

In all of the classes, the plural case markers are the same for all genders.

Class 1

The declension paradigm for class 1 goes as follows:

	singular			plural
	masculine	neuter	feminine	
nominative	*er*	*ys*	*y*	*y*
accusative	*a*	*ys*	*y*	*y*
dative	*um*	*um*	*er*	*a*

Table 11: Adjective Declension Class 1

Now, I could have simply matched each and every combination of gender, number and case with the appropriate ending – e.g. *+F+sg+acc:a*. But since the plural endings are the same for all genders, this would mean a lot of duplicated entries (three sets of completely identical endings).

Therefore, I first differentiated between the numbers by creating seperate continuation classes **adj_c1_sg** and **adj_c1_pl**.

The **adj_c1_pl** class then has a simple construction – first matching the gender markers with empty words, appending the plural marker and then matching the case markers with the endings.

adj_c1_pl	*+M:0*	**adj_c1_pl_ marker**	*+pl:0*	**adj_c1_pl_ case**	*+nom:^y*	#
	+N:0				*+acc:^y*	#
	+F:0				*+dat:^a*	#

Table 12: Adjective Class 1 Plural Lexicon

For the singular forms, seperating gender and number into their own lexicons would have resulted in a structure much more complex due to the overall goal to use common lexicons for common forms in the different adjective classes and the fact that the classes sometimes share common forms across different genders (f.e. a set of common suffixes for singular masculine, feminine and neuter dative in two classes. Here, it is much more sensible to pack match the three suffixes against the markers *+sg+M+dat*, *+sg+F+dat*, *+sg+N+dat* in one lexicon and to link to that lexicon from both of the two classes.)

The singular lexicon **adj_c1_sg** is structured in the following way, encompassing the subsets **adj_c1_c4_sg**, **adj_c1_2_3_4_sg**, **adj_c1_F** and **adj_c1_sg_dat_com** shared with other classes:

adj_c1_sg	**adj_c1_c4_sg**	**adj_c1_c2_c3_c4_sg**	*+M+sg+nom:^er*		#
			+M+sg+acc:^a		#
			adj_c1_F	*+F+sg+nom :^y*	#

				+F+sg+acc:^y #
		+N+sg+nom:^ys	#	
		+N+sg+acc:^ys	#	
	adj_c1_sg_dat_com	*+M+sg+dat:^um*	#	
		+F+sg+dat:^er	#	
		+N+sg+dat:^um	#	

Table 13: Adjective Class 1 Singular Lexicon

For example, it shares all of its forms except the dative endings with another class using the lexicon **adj_c1_c4_sg** while on the other side, sharing all the dative endings with a different class using **adj_c1_sg_dat_com**.

I could have even gone further by analysing underspecification between cases inside the the classes. For example, currently the lexicon **adj_c1_N** currently goes like this:

adj_c1_F	*+F+sg+nom:^y*	#
	+F+sg+acc:^y	#

Table 14: Adjective Class 1 Singular Feminine Endings

The endings for nominative and accusative fall together. So one could create a structure where the common ending is only stated once:

adj_c1_F	*+F+sg+nom:0*	**adj_c1_F_fictional**	*0:^y*	#
	+F+sg+acc:0			

Table 15: Adjective Class 1 Singular Feminine Endings Alternative Version

But for me, it was more important to analyse the parallels between the different classes since it also is of more grammatical relevance (remember the classes forming a continuum form strong to weak declension).

Class 2

Class 2 is identical with class 1, except that it introduces an alternative ending for singular neuter nominative and accusative: a suffix-less version. Both versions can occur in class 2. Therefore I relied on the previously established lexicons and just added a seperate lexicon for the two new forms **adj_c2_N** (also used by other classes):

+c2:0	**adj_c2**	**adj_c1_sg**	see above	
		adj_c2_N	*+N+sg* *+nom:0*	#
			+N+sg *+acc:0*	#
		adj_c1_pl	see above	

Table 16: Adjective Declension Class 1 Lexicon

Class 3

Class 3 is identical to class 2 – except for the neuter singular nominative and accusative, where only the endingless version added in class 2 can be found, but not the ending *ys* from class 1.

For the plural forms, class 3 forwards to **adj_c1_pl**. Due to the structure of class 1, class 3 needs to forward to **adj_c1_sg_dat_com** for the dative forms and to **adj_c1_c2_c3_c4_sg** for the other forms (neuter and masculine nominative and accusative) to exclude the neuter singular nominative and accusative forms (see above). Furthermore class 3 needs to forward to **adj_c2_N** for the special ending-less neuter singular nominative and accusative forms introduced in class two.

+c3:0	**adj_c3**	**adj_c3_c5_sg**	**adj_c2_N**	see above
			adj_c1_c2_c3_c4_sg	see above
		adj_c1_pl	see above	
		adj_c1_sg_dat_com	see above	

Table 17: Adjective Declension Class 3 Lexicon

Notice that **adj_c3** does not continue with **adj_c2_N** and **adj_c1_c2_c3_c4_sg** directly, but that it forwards to **adj_c3_c5_sg** which then continues witht he two lexicons. This is due to the fact that class 5 also uses this subset of class two neuter and class 1 masculine and feminine singular nominative and accusative endings.

Class 4

For the singular nominative and accusative forms, class 4 exhibits the same endings as class 1. Therefore it continues with **adj_c1_c4_sg**. For plural it also has the same forms as class 1, so it also has the option to continue with **adj_c1_pl**.

The differnece lies in the dative. Class 4 has two singular dative markers: *a* for all three genders, which is expressed by the lexicon **adj_c4_sg_dat_com** and shared with another lexicon, and the additional option *am* in the masculine and neuter:

<table>
<tr><td rowspan="7">+c4:0</td><td rowspan="7">adj_c4</td><td colspan="2">adj_c1_c4_sg</td><td colspan="4">see above</td></tr>
<tr><td>+M:0</td><td colspan="2" rowspan="2">adj_c4_sg_dat_MN</td><td rowspan="2">+sg+dat:^am</td><td rowspan="2">#</td></tr>
<tr><td>+N:0</td></tr>
<tr><td rowspan="3">adj_c4_sg_dat_com</td><td>+M:0</td><td rowspan="3">adj_c4_sg_dat_com2</td><td rowspan="3">+sg+dat:^a</td><td rowspan="3">#</td></tr>
<tr><td>+N:0</td></tr>
<tr><td>+F:0</td></tr>
<tr><td>adj_c1_pl</td><td colspan="4">see above</td></tr>
</table>

Table 18: Adjective Declension Class 4 Lexicon

Class 5

The declension pattern of class 5 comprises of already existing lexicons. The plural endings are identical to the previous classes (**adj_c1_pl**). The singular dative is built by appending *a* just as in class 4 (**adj_c4_sg_dat_com**). The singular nominative and accusative compose of the endings-less neuter markers introduced in class 2 and of the regular masculine and feminine markers introduced in class 1 – both reachable using the lexicon **adj_c3_c5_sg** established for class 3.

+c5:0	**adj_c5**	**adj_c1_pl**	see above
		adj_c4_sg_dat_com	see above
		adj_c3_c5_sg	see above
		+F+sg+nom:/oe	#
		+F+sg+acc:/oe	#
		+F+sg+dat:/oe	#

Table 19: Adjective Declension Class 5 Lexicon

However, as you can see, there are three additional alternative feminine singular endings all matching with */oe*. I introduced this multicharacter symbol – short for ‚optional ending' due to the fact that adjectives ending in *er* have an alternative, ending-less optional in the feminine singular for class 5. E.g. *kośer*:

	singular			plural
	masculine	neuter	feminine	
nominative	*kośer*	*kośer*	*kośery/kośer*	*kośery*
accusative	*kośera*	*kośer*	*kośery/kośer*	*kośery*
dative	*kośera*	*kośera*	*kośera/kośer*	*kośera*

Table 20: Adjective Declension Class 5

To cope with that, I could have introduced an alternative declension class for adjectives ending in *er* and declare it manually as a special continuation class for these adjectives just like the different types of noun declension classes. However, this observation is clearly regular and can be easily resolved with the help of a multicharacter symbol and a phonological rule:

PhonologicalRules.fst

```
...
> define Post2 ~[ $[ [ [ [ ? ? ] - {er} ] %/oe ] ] ] ;
> define Post Post1 .o. Post2 ;
…
> define Phon Phon1 .o. Phon2 .o. Phon3 .o. Phon4 .o.
      Phon5 .o.  Phon6 .o. Phon7a .o. Phon7b .o.
      Phon8 .o. Phon11 .o. Phon10 .o. Phon9 .o.
      Phon12 .o. Post ;
>
> define Elim [ {^} -> 0 , %/a -> 0 , %/oe -> 0 ] ;
>
> define PhonologicalRules Ablaut .o. Dat .o.
      SuppComp .o.    Phon .o. Elim ;
```

Rule **Post2** excludes all strings where the multicharacter symbol *oe* follows any two characters with the exception of *er*. Therefore constructions like *güt/oe*, i.e. adjectives that do not end in *er*, are prevented from getting this optional ending.

Then, by applying the rule **Elim**, */oe* gets matched with the empty word – since this alternative option is ending-less, as stated above.

Class 5a & 5b

Then, I had to add two additional classes, 5a and 5b, since the grammar defined two slight modifications of the class 5 scheme used in special constructions.

The first variant, class 5a, has an alternative ending for singular feminine dative: *y*. Therefore class 5a continues with either the class 5 lexicon **adj_c5** or matches the markers *+F+sg+dat* with the special ending *^y*.

+c5 a:0	**adj_c5a**	*+F+sg+dat:^y*	#
		adj_c 5	see above

Table 21: Adjective Declension Class 5a Lexicon

The second variant builds on the first one, only adding an alternative ending for the singular masculine nominative: *y*.

+c5 *b:0*	**adj_** **c5b**	*+M+sg+nom:^y*	*#*
		adj_ **c5a**	see above

Table 22: Adjective Declension Class 5b Lexicon

Class 6

Class 6 is probably the class with most number of modifications in the continuum.

	singular			plural
	masculine	neuter	feminine	
nominative	*y*	*y*	*y*	*a/y*
accusative	*a*	*y*	*y*	*a/y*
dative	*a*	*a*	*a/y*	*a*

Table 23: Adjective Declension Class 6

As you can see, it shares the same common singular dative suffix *a*, described by lexicon **adj_c4_sg_dat_com**. Furthermore, the feminine singular nominative and accusative endings *y* are identical to class 1 – described by lexicon **adj_c1_F**.

The plural suffixes share the same subset denoted by **adj_c1_pl** plus two new additional options: *y* for nominative and for accusative. Those, I had to denote manually in **adj_c6**.

The same goes for the suffixes for masculine and neuter nominative and accusative, as well as the alternative marker for feminine dative *y*.

+c6:0	**adj_c6**	**adj_c4_sg_dat_com**		see above	
		adj_c1_F			see above
		+M+pl:0	**adj_c4_pl2**	*+nom:^a*	#
		+F+pl:0		*+acc:^a*	#
		+N+pl:0			
		adj_c1_pl	see above		
		+M+sg+nom:^y	#		
		+M+sg+acc:^a	#		
		+N+sg+nom:^y	#		
		+N+sg+acc:^y	#		
		+F+sg+dat:^y	#		

Table 24: Adjective Declension Class 6 Lexicon

Adjectives in *-er*

When adjectives whose stem ends in *er* get appended with the suffix *-er*, only a single *er* is shown. The *er* does not double to **-erer*. This holds true independet of what the suffix *er* stands for. E.g. *kośer+Adjective+c5+M+sg+nom* → *kośer* and not **kośerer*, *kośer+Adjective+c3+F+sg+dat* → *kośer* and not **kośerer*.

Therefore I introduced a phonological rule matching the *^er* suffix with the empty word when it follows the string *er*:

PhonologicalRules.fst

```
...
> define Phon12 [ {^er} -> 0 || {er} _ ] ;
```

Adjectives in a

Adjectives ending in *a* are described to be a special case – being totally indeclinable – except for declension cass 4, where the masculine singular accusative and plural dative has an alternative form: *an*.

This change in declension pattern is clearly phonologically motivated and predictable. Therefore I did not want to manually mark the adjectives ending in *a* with a different continuation lexicon but rather let my algorithm append the correct endings. Due to simplicity I ended up using a filter system similar to the optional class 5 ending marked with the symbol */oe* which get filtered out when the stem does not end in *er* (see above).

This filter approach however, does come with a downside. Here, it means first generating two seperate declension tables and then sorting one out. It is not the most runtime efficient approach.

I first adjusted the lexicon **adj_marker** now having two options: 1. forwarding to the lexicon for regular suffixes by matching the *+Adjective* marker with the new multicharacter symbol */radj* (standing for ‚regular adjective') or 2. forwarding to the lexicon for the special *a*-adjectives by matching the adjective marker with the new multicharacter symbol */aadj* (standing for ‚a-adjective').

Grammar.txt

```
...
> LEXICON adj_marker
> +Adjective:/radj              adj_degree ;
> +Adjective:/aadj              adj_classes_a ;
```

The phonological rules then received a special set of pre-rules filtering out all combinations containing *a* followed by the marker */radj* and combinations containing a symbol except for *a* followed by the marker */aadj*. Then the remaining */radj* and */aadj* markers get deleted so that they do not interfere with the phonological rules happening at morpheme boundary.

PhonologicalRules.fst

```
...
> # Phonological Rules
> define Pre1 ~[ $[ [ [ ? - a ] %/aadj ] ] ] ;
> define Pre2 [ %/aadj -> 0 ] ;
> define Pre3 ~[ $[ [ a %/radj ] ] ] ;
> define Pre4 [ %/radj -> 0 ] ;
> define Pre Pre1 .o. Pre2 .o. Pre3 .o. Pre4 ;
…
> define Phon Pre .o. Phon1 .o. Phon2 .o. Phon3 .o.
      Phon4 .o.  Phon5 .o. Phon6 .o. Phon7a .o.
      Phon7b .o. Phon8 .o.  Phon11 .o. Phon10 .o.
      Phon9 .o. Phon12 .o. Post ;
...
```

Now, only the lexicon for the *a*-adjectives was missing. Since the ending is empty for all combinations of class, gender, number and case, I simply had to differentiate between the different markers. The only two distinc alternative endings were then simply listed in **adj_classes_a**.

adj_ marker *+Adject ive:/ aadj*	**adj_classes _a**	*+c1:0*	**adj_gen_a**	*+M:0*	**adj_num_ a**	*+pl:0*	***adj_case_ a***	*+nom: 0*	#
		+c2:0							
		+c3:0		*+N:0*				*+gen:0*	#
		+c4:0							
		+c5:0				*+sg:0*			
		+c5a: 0		*+F:0*				*+dat:0*	#
		+c5b: 0							
		+c6:0							
		+c4+M+sg+acc:^n							#
		+c4+M+pl+dat:^n							#

Table 25: a-Adjective Lexicon

Modification of Phon4 due to *er^a*-combination

Before I included adjectives into the system, the phonological rule 4 looked like this:

```
PhonologicalRules.fst        - Version 1 -
...
> define Phon4 [ r -> 0 || _ {^n} ,  _ {^a} , _
      {^s} ] ;
...
```

While this rule correctly converts the suffixed preposition *undyr^a* to **undy^a* which later is excluded by a filter, *kośer*^a (class 3 singular masculine accusative) is not supposed to be converted to **kośe^a* and then excluded. It is supposed to be matched correctly with *kośera*.

Therefore, I adjusted **Phon4** to exclude stems ending in *er* when appended with *^a*.

```
PhonologicalRules.fst      - Version 2 -
...
> define Phon4 [ r -> 0 || _ {^n} ,  [ ? - e ] _ {^a} ,
      _ {^s} ] ;
...
```

2.6.1 Comparatives

I used the marker *+comp* to mark the comparative. Instead of six (8), there are only three different declension classes in the comparative: 1. the indefinite class, used when the adjective precedes an indefinite article (*+indef*), 2. the definite class, used when the adjective precedes a definite article (*+def*), and 3. the predicative class, which comes into play when the adjective is used in a predicative manner (*+pred*).

The first two classes are actually very similar. They share a lot of common forms while only a few forms are specific to one of the classes. The third class, according to the GRAMMAR, contains all the forms in *er*. Since I the GRAMMAR did not include more information elaborating on this question (no declension table of the class), I included all forms containing the substring *er*. Furthermore, the Wymysorys literature I had available did not help to identify the correct forms either, because of the overall scarcity of adjectives in the comparative form.

In the following you can see a declension table of the three classes. Uncoloured cells contain an ending shared by both the definite and the indefinite class. Green cells are unique to the indefinite class. Blue cells are unique to the definite class. Underlined endings occur in the predicative class.

	singular									plural		
	masculine			neuter				feminine				
nominative	*erer*	*er*	*ery*	*er*	*erys*	*yś*	*ery*	*er*	*ery*	*er*	*ery*	*yn*
accusative	*era*		*yn*	*er*	*erys*	*yś*	*ery*	*er*	*ery*	*er*	*ery*	*yn*
dative	*era*		*yn*	*era*		*yn*		*era*	*yn*	*era*	*yn*	

Table 26: Adjective Comparatives Declension

Therefore I created a common lexicon for the indefinite and the definite class containg all the common endings: **adj_comp_com**. Forms that also occur in the predicative class were

further extracted into separate lexicons that the predicative class could then later forward to.

In **adj_comp_com**, the subset **adj_comp_com_pl** deals with all the plural forms – sequentially concatenating the gender markers, the plural marker and then the case markers – all matching with the ending *yn*.

Additionally, a second route takes the algorithm to a seperate lexicon after the plural marker: to **adj_comp_com_er_pl_case**. As described above, this lexicon contains the plural forms containing *er* (hence the name) which are also used in the predicative class: *ery* for nominative and accusative and *era* for dative.

<table>
<tr><td rowspan="6">adj_comp_com_pl</td><td>+M:0</td><td rowspan="6">adj_comp_com_pl2</td><td rowspan="6">+pl:0</td><td rowspan="6">adj_comp_com_pl3</td><td>+nom:0</td><td rowspan="3">adj_comp_com_pl_underspec</td><td rowspan="3">0:^yn</td><td rowspan="3">#</td></tr>
<tr><td>+F:0</td><td>+acc:0</td></tr>
<tr><td>+N:0</td><td>+dat:0</td></tr>
<tr><td colspan="2" rowspan="3">adj_comp_com_er_pl_case</td><td>+nom:^ery</td><td>#</td></tr>
<tr><td>+acc:^ery</td><td>#</td></tr>
<tr><td>+dat:^era</td><td>#</td></tr>
</table>

Table 27: Adjective Comparative Common Plural Lexicon

For the singular I observed that the dative endings actually all fall together independent of gender. Therefore I created the lexicon **adj_comp_com_dat** that matches the *yn*-ending with the dative marker and also forwards to an additional lexicon **adj_comp_com_er_dat** containing the second dative marker *era* also found in the predicative class.

Furthermore, the neuter and the feminine gender have even more similarities: The nominatice and accusative ending *ery*. Since these two forms also occur in the predicative class, I put them into a seperate lexicon: **adj_comp_com_er_FN**.

The common lexicon **adj_comp_com_FN** is now the continuation after appending the neuter or feminine markers. It forwards to **adj_comp_com_er_FN** or to **adj_comp_com_dat** for the dative markers. The feminine gender has also a second option **adj_comp_com_er_F**, since it has the alternative endings *ery*, as seen obove.

The masculine gender has got three distinct endings that are also part von the predicative class and therefore put into the sepcial lexicon **adj_comp_com_er_M**: *er* and *erer* for nominative and *era* for dative. Furthermore it has the ending *yn* for accusative which is not part of **adj_comp_er_M**.

<table>
<tr><td rowspan="6">+M+sg:0</td><td rowspan="6">adj_comp_com_M</td><td>+acc:^yn</td><td>#</td><td colspan="3"></td></tr>
<tr><td rowspan="3">adj_comp_com_er_M</td><td>+nom:^er</td><td>#</td><td colspan="3"></td></tr>
<tr><td>+nom:^erer</td><td>#</td><td colspan="3"></td></tr>
<tr><td>+acc:^era</td><td>#</td><td colspan="3"></td></tr>
<tr><td colspan="2" rowspan="2">adj_comp_com_dat</td><td>+dat:^yn</td><td>#</td><td colspan="3"></td></tr>
<tr><td>adj_comp_com_er_dat</td><td>+dat:^era</td><td>#</td><td></td></tr>
<tr><td rowspan="5">+F+sg:0</td><td rowspan="5">adj_comp_com_F</td><td rowspan="3">adj_comp_com_FN</td><td rowspan="2">adj_comp_com_er_FN</td><td>+nom:^er</td><td>#</td><td colspan="3"></td></tr>
<tr><td>+acc:^er</td><td>#</td><td colspan="3"></td></tr>
<tr><td>adj_comp_com_dat</td><td colspan="5">see above</td></tr>
<tr><td rowspan="2">adj_comp_com_er_F</td><td>+nom:^ery</td><td colspan="5">#</td></tr>
<tr><td>+acc:^ery</td><td colspan="5">#</td></tr>
<tr><td>+N+sg:0</td><td>adj_comp_com_N</td><td>adj_comp_com_FN</td><td colspan="6">see above</td></tr>
</table>

Table 28: Adjective Comparative Common Singular Lexicon

Indefinite Class

The described plural and singular endings are all reached by the **adj_comp_com** lexicon that is used for the indefinite (**adj_comp_indef**) and the definite (**adj_comp_def**) classes. Apart from this, as already explained, both classes also have a hand full of distinct endings to them. Therefore **adj_comp_indef** also continues with the following set of lexicons, containing four seperate endings in the neuter gender – two of which are shared by the predicative class and therefore put into a seperate sublexicon - and two special plural suffixes also found in the predicative class, bunched together in the sublexicon **adj_comp_indef_er_pl**.

+N+sg:0	**adj_comp_indef_N**	**adj_comp_indef_er_N**	*+nom:^erys*	#			
			+acc:^erys	#			
		+nom:^yś	#				
		+acc:^yś	#				
adj_comp_indef_er_pl		*+M:0*	**adj_comp_indef_er_pl2**	*+pl:0*	**adj_comp_indef_er_pl3**	*+nom:^er*	#
		+F:0				*+acc:^er*	#

Table 29: Adjective Comparative Indefinite Lexicon

Definite Class

The definite lexicon is very simply structured: it only contains a link to the common **adj_comp_com** lexicon and three additional endings not found in the indefinite class (but in the predicative class!):

+def:0	**adj_comp_def**	**adj_comp_com**	see above		
		+N+sg:0	**adj_comp_er_def_N**	*+nom:^ery*	#
				+acc:^ery	#
		+M+sg:0	**adj_comp_er_def_M**	*+nom:^ery*	#

Table 30: Adjective Comparative Definite Lexicon

Predicative Class

Since the third class actually only consists of all the previous forms in *er*, I now just had to correctly list the lexicons I had already named with the component **_er_** e.g. **adj_comp_er_def_F**. Here is the full structure diagram for all comparative classes with the predicative class at the bottom:

adj_comp	*+indef:0*	**adj_comp_indef**	**adj_comp_com**	*+M+sg:0*	**adj_comp_com_M**	*+acc:^yn*	#									
						adj_comp_com_er_M	*+nom:^er*	#								
							+nom:^erer	#								
							+acc:^era	#								
						adj_comp_com_dat			*+dat:^yn*	#						
									adj_comp_com_er_dat	*+dat:^era*	#					
				+F+sg:0	**adj_comp_com_F**	**adj_comp_com_FN**	**adj_comp_com_er_FN**		*+nom:^er*	#						
									+acc:^er	#						
							adj_comp_com_dat		see above							
						adj_comp_com_er_F	*+nom:^ery*		#							
							+acc:^ery		#							
				+N+sg:0	**adj_comp_com_N**	**adj_comp_com_FN**	see above									
				adj_comp_com_pl		*+M:0*	**adj_comp_com_pl2**	*+pl:0*	**adj_comp_com_pl3**	*+nom:0*	**adj_comp_com_pl_understec**	*0:^yn*	#			
						+F:0				*+acc:0*						
						+N:0				*+dat:0*						
										adj_comp_com_er_pl_case		*+nom:^ery*	#			
												+acc:^ery	#			
												+dat:^era	#			

			+N+sg:0	**adj_comp_indef_N**	**adj_comp_indef_er_N**	*+nom:^erys*	#					
						+acc:^erys	#					
					+nom:^yś	#						
					+acc:^yś	#						
			adj_comp_indef_er_pl		*+M:0*	**adj_comp_indef_er_pl2**	*+pl:0*	**adj_comp_indef_er_pl3**	*+nom:^er*	#		
					+F:0				*+acc:^er*	#		
					+N:0							
	+def:0	**adj_comp_def**	**adj_comp_com**	see above								
			+N+sg:0	**adj_comp_er_def_N**		*+nom:^ery*	#					
						+acc:^ery	#					
			+M+sg:0	**adj_comp_er_def_M**		*+nom:^ery*	#					
	+pred:0	**adj_comp_pred**	*+M+sg:0*	**adj_comp_pred_M_sg**	**adj_comp_com_er_M**	see above						
					adj_comp_er_def_M	see above						
					adj_comp_com_er_dat	see above						
			+F+sg:0	**adj_comp_pred_F_sg**	**adj_comp_com_er_FN**	see above						
					adj_comp_com_er_dat	see above						
					adj_comp_com_er_F	see above						
			+N+sg:0	**adj_comp_pred_N_sg**	**adj_comp_com_er_FN**	see above						
					adj_comp_com_er_dat	see above						
					adj_comp_indef_er_N	see above						

					adj_comp_er_def_N	see above
			+M+pl:0	**adj_comp_com_er_pl_case**		see above
			+F+pl:0			
			+N+pl:0			
			adj_comp_indef_er_pl	see above		

Table 31: Adjective Comparative Lexicon

Comparative for a-Adjectives

Unfortunately, no indication was given by the GRAMMAR as to what the comparative forms of *a*-Adjectives are.

Occurence of Vowel Change in Comparative form

A lot of adjectives change their root vowel in the comparative form – similar to the Ablaut-feature of nouns. Therefore I included the same multicharacter marking (e.g. */u*, */2u*, …, see the section about nouns for more information) in front of vowels object to change just as I did with the nouns and added the */d* marker to match against the *+comp*-marker to trigger the vowel changes:

```
Grammar.txt
...
> LEXICON adj_degree
>                       adj_classes ;
> +comp:/comp/d         adj_comp ;
…
```

See the next section for an explanation for the */comp*-symbol.

Suppletive roots

Some adjectives change the stem when changed into comparative form: *güt* ‚good' → *beser*, not **güter*, *ryś* ‚early' → *ejer*, not **ryśer*. In order to avoid introducing more lexicons in the grammar and declaring the roots twice, I came up with another solution:

Grammar.txt

```
...
> LEXICON adj_degree
>                     adj_classes ;
> +comp:/comp/d       adj_comp ;
…
```

The comparative marker *+comp* gets matched against the multicharacter symbol */comp* in the lexicon. A set of phonological rules then check for the existance of this symbol in combination with the roots *güt* and *ryś* and replaces them with their counterparts. The */comp*-marker is then removed.

PhonologicalRules.fst

```
...
> # Suppletive Comparative
> define SuppComp1 [ {güt} -> {bes} , {ryś} -> {ej} ||
      _ ?* %/comp ] ;
> define SuppCompElim [ %/comp -> 0 ] ;
> define SuppComp SuppComp1 .o. SuppCompElim ;
…
> define PhonologicalRules Ablaut .o. Dat .o.
      SuppComp .o. Phon .o. Elim ;
```

Irregularity: jer-form

Various adjectives have comparative forms slightly different to what xfst put out using the rules as of this point: *łengjer* instead of **łenger*, *jyngjer* instead of **jynger*, *hyhjer* instead of **hyher* and *śtàkjer* instead of **śtàker*. It seems as if there is an insertion of *j* in the cases where *g*, *h* or *k* get followed by a morpheme boundary and an *e*. This called for the modification of the **Phon7b** rule, currently describin the same phenomenon for the cases when *h* or *k* get followed by an *a*.

Notice that the combination *g^a* does not change to **gja* (e.g. *oüg* 'eye' → *oüga* instaed of **oügja*, plural nominative). But the combination *g^e* does indeed seem to change to *gje* as can be seen by the preceding examples.

Therefore I modified rule **Phon7b** to include the three mentioned patterns and match the morpheme boundary with a *j*:

PhonologicalRules.fst

```
...
> define Phon7b [ %^ -> j || [ k | h | V - y ] _ a ,
      [ k | g | h ] _ e ] ;
```

Unfortunately, the algorithm had a problem recognizing the superlative forms. So I checked by inputting a lower form using down ołd+Adjective+sup+c6+N+pl+dat and got **ełdyśta* as an output. This was clearly wrong! *^st^a* was substituted with *yśta*. The culprit was certainly a phonological rule.

As you may remember, rule **Phon11** inserts an *y* into a morpheme boundary when a consonant and an *n* or an *s* come together. This was very useful for genitive noun patterns but problematic in this case. Here, **Phon11** changes *ełd^st^a* to *ełdy^st^a*. Then, rule **Phon9** recognizes the pattern *y^s* and replaces it with *yś* creating *ełdyśt^a*.

So, apparently, the described process should take place for a single appended *s* but not for cases of longer affixation like *st*.

To fix this issue, I limited **Phon11** to cases where the *s* gets followed by the ending symbol *.#.*, i.e. where it is the last character:

PhonologicalRules.fst

```
...
> define Phon11 [ %^ -> y %^ || [ C - l - ł - f ] _ [ n
      | s .#. ] ] ;
```

Now, superlative adjectives were not longer ill-formed. But now, unfortunately, genitive words like *kynd^s (→ kyndyś)* also did no longer undergo the modifications described in **Phon11** and **Phon9** even though *s* seems to be the last symbol of the string.

The reason was that the lexicon appends the Flag Diacritic *@R.GEN.ALLOWED@* after the genitive ending, which is not visible by the user but still treated as a multicharacter symbol by the xfst. There is, however, a setting named flag-is-epsilon described in THE BOOK (p. 362) which, when activated, makes the xfst composition algorithm treat Flag Diacritics as epsilons.

Since the Flag Diacritic prevents **Phon11** from processing the single *s*-ending correctly by recognizing following characters, I added the declaration flag-is-epsilon = ON to the Main.fst script:

Main.fst

```
...
> regex PhonologicalRules ;
>
> read lexc < Grammar.txt ;
>
> set flag-is-epsilon ON
...
```

Adjectives ending in s or c

Adjectives that have a stem ending in *s* or *c* needed an additional rule. Observations showed that they receive the ending *ty rather than* sty*: grus* 'big' → *grysty*, not **gryssty*, *küc* 'short' → *kjycty*, not **kjycsty.*

This could be solved by adding a phonological rule, eliminating the *s* from the ending when it gets preceded by an *s* or *c* and a morpheme boundary:

PhonologicalRules.fst

```
...
> define Phon13 [ s -> 0 || [ s | c ] %^ _ ] ;
...
```

Irregularity: gröp → grywer

The Adjective *gröp* ‚fat' displays an unusual change to *grywer* in the comparative form. The vowel change can be explained by the Ablaut-feature. The apparent change from *p* to *w* in the stem hower, cannot. Therefore, for lack of observations similar to this which would indicate a productive rule, I declared *grywer* as the suppletive comparative root for *gröp* in the way I did it for *güt/beser* in the previous section.

2.6.2 Superlatives

Superlative forms are derived by appending the suffix *st* to the adjective and then inflecting it in the regular adjectival pattern. Therefore, I just matched the superlative marker *+sup* with the suffix *st* in the **adj_degree** lexicon and declared the **adj_classes** lexicon (regular class selection) as the continuation of this path.

Occurence of Vowel Change & Suppletive Roots in Comparative form

Similar to the comparatives, the superlative forms also exhibit change of vowel (Ablaut) and suppletive roots. The rules and patterns seem to be exactly the same as for the comparatives. Therefore, I just added the two multicharacter symbols */comp* and */d* to match with the superlative marker. Please refer to the previous section for information on how they work.

Grammar.txt

```
...
> LEXICON adj_degree
>                          adj_classes ;
> +comp:/comp/d            adj_comp ;
> +sup:/comp/d^st          adj_classes ;
```

Superlative for a-Adjectives

Unfortunately, no indication was given by the GRAMMAR as to what the superlative forms of *a*-Adjectives are.

3 Testsuit

3.1 Perform analyses

Similar to a normal xfst prompt you can perform analyses using the commands up <string> and down <string>. Although this can be a pretty helpful feature, it is not recommended when regarding perfomance. Each command call lets the python script create a new subprocess, call the xfst program and compose the Wymysorys grammar again.

The command uplist <word1> <word2> … lets you analyse several words at once. The words need to be seperated by spaces.

The command inflect <word> <type> prints out all possible inflected forms of a certain word. The argument <word> needs to be inserted with the base form of the word used in the upper form. The following options are possible for <type>:

- Article
- Preposition
- Noun
- Adjective

The possible categories for each word type get imported from the file WordTypes/Types.txt. Notice that the output is not well structured and that it takes a while to compute the forms due to redundant processes (see xfstlib.py for more insight). A future version of the program would need to output the data as a structured table.

inflect <word> <type> -existing limits the output to existing word forms.

3.2 Testing

3.2.1 Perfoming a test

The test suit provides an easy way for automated testing. You can type in the command test <category> <direction> to perform a test. <category> denotes the type of words that you want to be tested. The following options are currently available: Adjective, Article, Noun, Preposition. <direction> needs to be replaced with either up if one wants the test script to take lower word forms as input and test them against proper upper word analyses or down if one wants the test script to take upper forms as input and test them against proper lower word forms.

The command testall performs all test combinations in a row.

`Case 1 – no error`

Performing the command might and should look like this:

```
> (WymysorysL): test Noun l
> END OF TEST, predicted 50 entries correctly
> Predicted 0 falsely
```

If this is the output given by the test suit it means that all of the given test entries have been analysed corretly by the xfst script matching the supposed output.

`Case 2 – False prediction`

If the script predicts an analysis that should not be predicted the output may look like this:

```
> (WymysorysL): test Article l
> ERROR!
> Falsely predicted: der+Article+def+N+sg+acc at entry 1
> END OF TEST, predicted 6 entries correctly
> Predicted 1 falsely
```

Each error is reported seperately with information regarding the problematic word (e.g. at entry 1). The section *Explaining the data* gives more information about where to look for that word.

Case 3 – Prediction missing

If the script fails to predict a certain analysis the output may look like this:

```
> (WymysorysL): test Article l
> ERROR!
> Not predicted: der+Article+def+N+sg+dat at entry 1
> END OF TEST, predicted 6 entries correctly
> Predicted 1 falsely
```

The last two lines tell you how many input strings were analysed entirely correctly (predicted 6 entries correctly) and how many input strings contain at least one analysis error (Predicted 1 falsely).

3.2.2 Explaining the data

Several types of test data is available in the test suit. The test data is given in .txt format and located in /Wymysorys/TestData/

The test data is split into two categories: the lower sides of words (e.g. *Adjective_l.txt*) and the upper sides of words (e.g. *Noun_u.txt*). The files contain several word forms seperated by an empty line.

For each test file there is a seperate file containing the proper analysis. They are named systematically (<test file name>*_correct.txt* e.g. *Noun_l_correct.txt*). They follow the structure of the test files listing the analyses seperated by empty lines. If there are several analyses for one input form they are given underneath each other without empty lines seperating them. Here is an example with “>” denoting a line.

Noun_l.txt

```
…
>
> pyś
>
> kyz
>
…
```

Noun_l_correct.txt

```
…
>
> puś+Noun+M+pl+nom
> puś+Noun+M+pl+acc
>
> kyz+Noun+M+pl+nom
> kyz+Noun+M+pl+acc
> kyz+Noun+M+sg+nom
> kyz+Noun+M+sg+acc
> kyz+Noun+M+sg+dat
>
…
```

The following test data is available under /Wymysorys/TestData/

Category	**Input data**	**Correct analyses**
Adjective	Adjective_l.txt	Adjective_l_correct.txt
	Adjective_u.txt	Adjective_u_correct.txt
Article	Article_l.txt	Article_l_correct.txt
	Article_u.txt	Article_u_correct.txt
Noun	Noun_l.txt	Noun_l_correct.txt
	Noun_u.txt	Noun_u_correct.txt
Preposition	Preposition_l.txt	Preposition_l_correct.txt
	Preposition_u.txt	Preposition_u_correct.txt

Table 32: Table of Test Data

The addition of more test data can be easily perfomed by modifiying the listed files. Adding a test file for a new word type can also be performed without much struggle by creating four seperate text files according to the naming scheme.

3.3 Quit

The test suit can be turned off by typing the command quit.

3.4 Utilities

There are several utility commands that can be of help when using the test suit:

<table>
<tr><th>command</th><th>Arguments</th><th>Function</th><th>Annotations</th></tr>
<tr><td rowspan="3">l</td><td></td><td>Calls the command most recently performed by the user.</td><td rowspan="3">Calling this command does not add it to the list of perfomed commands.</td></tr>
<tr><td><index></td><td>Calls the command performed by the user as many steps before as provided by <index>.
e. g.
<code>l 1</code> perfoms the most recent command
<code>l 2</code> perfoms the second most recent command</td></tr>
<tr><td>s</td><td>Gives the list of all previously called commands.</td></tr>
<tr><td>debug</td><td>[false, true]</td><td>Activates debug mode. Gives more detailed output to the console when interacting with the program. Performing the command without an</td><td>Calling this command does not add it to the list of perfomed commands.</td></tr>
</table>

		argument toggels the debug mode.	

Table 33: Utility Functions

4 The goal

My goal was to model a system that could take a sentence from my physical copy of The little prince in Wymysorys (“Der Kliny Fjyśt”) and output a list of possible analyses for all of the words in the sentence which could later be used for different language applications.

Unfortunately, while I succeeded modelling several word types and very interesting language phenomena like Ablaut, I did not succeed in reaching my initial goal. This is mainly due to the fact that I did not investigate verb inflection. Therefore, fully annotating a grammatically correct sentence is not possible yet.

But in theory, the application should be ready to accept simple noun phrases. All the prerequisites are there: articles, adjectives and nouns.

1. First test: Der Kliny Fjyśt

The command uplist allows to input several words into the test suit. They must be in lower case and stripped of any punctuation.

I started with the title of the book:

```
> (WymysorysL): uplist der kliny fjyśt
> der+Article+def+M+sg+nom
> der+Article+def+N+pl+gen
> der+Article+def+F+pl+gen
> der+Article+def+M+pl+gen
>
> klin+Adjective+c6+M+pl+nom
> klin+Adjective+c6+M+pl+acc
> klin+Adjective+c6+M+sg+nom
> klin+Adjective+c6+F+pl+nom
> klin+Adjective+c6+F+pl+acc
> klin+Adjective+c6+F+sg+nom
> klin+Adjective+c6+F+sg+acc
> klin+Adjective+c6+F+sg+dat
> klin+Adjective+c6+N+pl+nom
...
> klin+Adjective+c1+M+pl+nom
> klin+Adjective+c1+M+pl+acc
> klin+Adjective+c1+N+pl+nom
> klin+Adjective+c1+N+pl+acc
> klin+Adjective+c1+F+pl+nom
> klin+Adjective+c1+F+pl+acc
> klin+Adjective+c1+F+sg+acc
> klin+Adjective+c1+F+sg+nom
>
> fjyśt+Noun+M+sg+acc
> fjyśt+Noun+M+sg+nom
```

Well, that worked flawlessly. Notice the large number of possible forms for the adjective *kliny* – 71 in total. The adjectival classes indeed share a lot of common endings!

As you can see by the forms marked in yellow, the correct, congruent analyses can be found as expected.

2. Second step: Myta obrozła fum oütor

Next, I put in the second expression found in the book. It translates to: ‘with the paintings from the author’. The polish loanword *obrazła* ‘painting’ is an expample for why the language is not mutually intelligible with German.

Notice that it contains two suffixed prepositions. I did not expect them to be of use so early.

```
> (WymysorysL): uplist myta obrozła fum oütor
> myt+Preposition+Cl+Article+def+M+sg+acc
> myt+Preposition+Cl+Article+def+M+pl+dat
> myt+Preposition+Cl+Article+def+F+pl+dat
> myt+Preposition+Cl+Article+def+N+pl+dat
>
> obrozł+Noun+M+sg+acc
> obrozł+Noun+M+sg+dat
> obrozł+Noun+M+pl+nom
> obrozł+Noun+M+pl+acc
> obrozł+Noun+M+pl+dat
>
> fu+Preposition+Cl+Article+def+M+sg+dat
> fu+Preposition+Cl+Article+def+N+sg+dat
>
> oütor+Noun+M+sg+nom
> oütor+Noun+M+sg+acc
> oütor+Noun+M+sg+dat
>
```

You can see that the second test did also work. The four correct dative forms are among the four analyses.

3. The limitations: Ym projekt błyn dy obrozła fum oütor gybroüht

Finally, I used a proper sentence, I found on one of the first pages, as input. It translates to "In the project the paintings from the author remained used." The word *błyn*, 3rd person plural preterite of *błąjn* 'remain, become', is another example for the an unintelligible word for German speakers – and polish speakers as well – in contrast to *obrozła,* which polish speakers can understand. Therefore, it does not seem to be sufficient to speak both German and Polish to understand Wymysorys. I, myself, as a native speaker of both Polish and German, can at least make that statement.

```
> (WymysorysL): uplist ym projekt błyn dy obrozła fum oütor
                                                    gybroüht
> yn+Preposition+Cl+Article+def+M+sg+dat
> yn+Preposition+Cl+Article+def+N+sg+dat
> der+Article+def+M+sg+dat
> der+Article+def+N+sg+dat
>
> projekt+Noun+N+sg+dat
> projekt+Noun+N+sg+acc
> projekt+Noun+N+sg+nom
>
> ???
>
> der+Article+def+F+sg+acc
> der+Article+def+F+sg+nom
> der+Article+def+N+pl+acc
> der+Article+def+N+pl+nom
> der+Article+def+F+pl+acc
> der+Article+def+F+pl+nom
> der+Article+def+M+pl+acc
> der+Article+def+M+pl+nom
>
> obrozł+Noun+M+sg+acc
> obrozł+Noun+M+sg+dat
> obrozł+Noun+M+pl+nom
> obrozł+Noun+M+pl+acc
> obrozł+Noun+M+pl+dat
>
> fu+Preposition+Cl+Article+def+M+sg+dat
> fu+Preposition+Cl+Article+def+N+sg+dat
>
> oütor+Noun+M+sg+nom
> oütor+Noun+M+sg+acc
> oütor+Noun+M+sg+dat
>
> ???
>
```

Obviously the script can only output a partial analysis. Still, it does its work as expected. The previous noun phrase *dy obrozła fum oütor* is now in the nominative case since it is the subject of a passive construction. These forms are also correctly found in the analysis.

5 The resume

Working on Wymysorys was certainly not an easy task. There is not much documentation about the language that I could use in my work. The available material, while certainly very helpful work, is still only a small step into fully understanding the complexity of the language. This, given the small number of Wymysorys speakers, the GRAMMAR and work by Tymoteusz Król, the leading figure in the revitalization movement, is definetely remarkable.

It is still going to be a long way for _____. And even longer for actual results in the field of computational linguistics regarding the Wymysorys language – depending on whether more people will consider it valuable to put their work into small languages that are on the verge of dying. While this certainly will not bring along many profits, maybe it can help to prevent the loss of history, culture and linguistic heritage.

6 Sources

Andrason, Alexander & Król Tymoteusz. (2016). *A Grammar of Wymysor*

Hiermit erkläre ich an Eides statt, dass ich die Haus-/Bacherlorarbeit mit dem Titel --- selbstständig verfasst und keine anderen als die angegebenen Quellen verwendet habe. Die Stellen der Arbeit sowie evtl. beigefügte Abbildungen, Zeichnungen oder Grafi-ken, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, habe ich unter Angabe der Quelle als Entlehnung kenntlich gemacht. Ort, Datum Unterschrift