UPPSALA UNIVERSITY

Project ref. no.	LE3-4239
Project title	***SCARRIE Scandinavian Proof-reading Tools***
Deliverable number	*DEL 3.4.1*
Deliverable title	***A Swedish Hyphenation Marker***
Number of pages	*38*

WP/Task responsible	*Anna Sågvall Hein, Department of Linguistics, Uppsala University, Box 513, S-751 20 Uppsala, Sweden* *Email: anna@ling.uu.se*
Author(s)	*Leif-Jöran Olsson,* ljo@stp.ling.uu.se
EC Project Officer	*Pierre-Paul Sondag*

Keywords	*Hyphenation marker, hyphenation, morphological hyphenation*

Abstract *As a part of the SCARRIE-project, this report outlines the addition of hyphenation rules to the Uppsala Chart Processor (UCP) grammar. Prerequisites were to follow the morphological hyphenation principle where possible and where not possible the mechanical hyphenation principle were to be used.*

EXECUTIVE SUMMARY

Since hyphenation errors are so common, [WED98], it is very important to implement hyphenation in the SCARRIE proof-reading tools. The hyphenation must be at word internal places where it does not interrupt the reader.

The morphological hyphenation is the main method used in this implementation.

The hyphenation rules have been added to nouns, verbs, adjectives, adverbs and compounds. Other word classes do not have any inflection fulfilling the rules for morphological hyphenation.

The mechanical hyphenation has not been implemented.

Uppsala University
Department of Linguistics
SCARRIE
Leif-Jöran Olsson, ljo@stp.ling.uu.se

1997-12-06

Addition of Hyphenation rules to the Uppsala Chart Processor Grammar

Abstract:
As a part of the SCARRIE-project, this report outlines the addition of hyphenation rules to the Uppsala Chart Processor (UCP) grammar. Prerequisites were to follow the morphological hyphenation principle where possible and where not possible the mechanical hyphenation principle were to be used.

1. INTRODUCTION 5

1.1 Morphological Hyphenation 5
1.2 Mechanical Hyphenation 5

2. NOUNS 6

2.1 Preparing Hyphenation/Syllabification of Nouns 6
2.2 Putting the Noun-morphs together 7
2.2.1 nomen.avstavning 8
2.3 Changes to Noun Stems 10
2.4 Stem-internal Hyphenation 11
2.5 Noun Compounds 11
2.5.1 comp.avstavning 12

3. VERBS 14

3.1 Preparing Hyphenation/Syllabification of Verbs 14
3.1.1 The Attribute same 15
3.1.2 The Attribute linkb 16
3.2 Putting the Verb-morphs together 16
3.2.1 verb.avstavning 17
3.3 Changes to Verb Stems 18

4. ADJECTIVES 19

4.1 Preparing Hyphenation/Syllabification of Adjectives 19
4.1.1 The Attribute same 20
4.1.2 The Attribute linkb 21
4.2 Putting the Adjective-morphs together 21
4.2.1 adj.avstavning 22
4.3 Changes to Adjective Stems 23

5. ADVERBS 23

5.1 Preparing Hyphenation/Syllabification of Adverbs 23
5.1.1 The Attribute same 24
5.1.2 The Attribute linkb 25
5.2 Putting the Adverb-morphs together 25
5.2.1 adverb.avstavning 25
5.3 Changes to Adverb Stems 26

6. A FEW WORDS ABOUT THE FUTURE 26

REFERENCES 26

APPENDIX A: TEST NOUNS AND THEIR HYPHENATIONS 26

APPENDIX B: TEST VERBS AND THEIR HYPHENATIONS 32

APPENDIX C: TEST ADJECTIVES AND THEIR HYPHENATIONS 36

APPENDIX D: TEST ADVERBS AND THEIR HYPHENATIONS 38

1. Introduction

Hyphenation errors are among the most frequent errors in the newspaper texts studied in the Swedish part of the SCARRIE-project [WED98]. A correct hyphenation is essential for a fluent reading. In order to add morphological hyphenation-rules to the Uppsala Chart Processor (UCP) grammar there are three cases to think of:

1. Affixes, mostly suffixes
2. Stem-internal
3. When to use Mechanical hyphenation

1.1 Morphological Hyphenation

The morphological hyphenation-rules used follow the recommendations of The Swedish Board of Language in "Svenska skrivregler" [1], with the following exceptions as expressed in "Svensk avstavningsordlista" [2]:

1. The syllables have to be pronounceable.
2. The sje-sound -ge must not be hyphenated between g and e.
3. Double-m before suffixes of inflected forms is hyphenated between the two m's.
4. The letter c is not allowed immediately before a hyphen, since no Swedish word ends in c.
5. There have to be at least two letters on both sides of the hyphen.

1.2 Mechanical Hyphenation

Mechanical hyphenation is much simpler than morphological hyphenation, since its essence is break before CV. i.e. ‘fil-mer’ compared to morphological ‘film-er’. For a proper definition, see [1]. The mechanical hyphenation has been decided to be handled by an external program, since it would be hard to implement it effectively in UCP.

2. Nouns

In this chapter the addition of hyphenation-rules to the noun-rules in the UCP-grammar will be outlined. The changes have mainly been done to the inflectional pattern-rules and the core noun-rules have been changed as little as possible. The basic idea has been to interact with the morphological analysis, so that the overhead would be as small as possible.

2.1 Preparing Hyphenation/Syllabification of Nouns

In order to get at the suffixes, for hyphenation, the letters have to be saved during the dictionary consultation. Otherwise the letters of the suffixes are lost. (This is also true for the stem.). The way to do it is the same for all dictionaries affected; *stem* (sve.dic), *plur* and *form*. Since the morphs must be longer than one letter, according to rule five in the exceptions to the morphological hyphenation-rule outlined in section 1.1, this excludes the case dictionary.
In the definition of a dictionary there is an attribute called gathering-rule, which can be assigned a rule-body to do this collection of letters. In the following example gathering-rule unifies the letters of the stem with the attribute :last by concatenation. (See Fig. 2.1.)

```
(define dictionary sve.dic
             (search-key #! char;)
             (forward-action #u advance;)
             (gathering-rule #u
                          (<& 1>,<&  :new>:=:<& :last><* char>
                          / <& :new>:=:<* char>);)
)
```

Fig. 2.1: Dictionary definition with the attribute gathering-rule containing a rule-body for letter collection.

The value of the attribute :last has to be assigned to another attribute before the consultation of another suffix-dictionary otherwise it will be lost. This is done in the following rules:

```
(define sve.gram-entry plur.suff
 #u           (<* morph.cat> = 'numb.suff,
              <& numb> :=: 'plur,
              <& decl> :=: <* decl>,
              (<& plurmorf> / <& plurmorf> :=: <* :last>),
              not <& plurmorf> = 'nil,
              (<* eow>,
              <& form> :=: <* form>,
              <& case>:=:<* case>,
              close.noun/advance(form.suff))
              / <* morph.cat> = 'form.suff,
              (<& decl> = '- / <&  numb> :=: 'sing),
              form.suff
              / <* morph.cat> = 'case.suff,
              (<& decl> = '- / <& numb> :=: 'sing),
              <& form>:=:'indef,case.suff);
 #! morph.cat;
)

(define sve.gram-entry form.suff
   #u         (<* morph.cat>='form.suff,
              <&  form>:=:<* form>,
              <&  gender>:=:<* gender>,
              <&  numb>:=:<* numb>,
              (<& formmorf> / <& formmorf> :=: <* :last>),
              not <& formmorf> = 'nil,
              (<* eow>,
              <& case> :=: <* case>,
              close.noun
              / advance,
              case.suff)
              / case.suff);
)
```

Fig. 2.2: Unification of suffix-morphs with the attributes plurmorf and formmorf (in italics).

The stem is assigned to the attribute dic.stem. For a deeper understanding of the UCP-formalism please consult [3] or other related papers.

2.2 Putting the Noun-morphs together

With the noun-morphs available in the attributes stem, which is an elaboration on dic.stem, plurmorf and formmorf, they have to be put together with an appropriate separator in between them. Underscore, _, was chosen as the separator, since it was not present in the newspaper-corpora analysed, but this can be changed if needed.
To cover all possible ways to end parsing of nouns a call to the new rule nomen.avstavning (noun hyphenation) was added to the rules in Fig. 2.3:

a.
```
(define sve.gram-entry store.noun
 #u           nomen.avstavning,
              assign.majorprocess('np_noun),
              store.single.morph;
)
```

b.
```
(define sve.gram-entry close.noun
 #u           (<& numb> / <& numb> :=: <& numb>),
              (<& form> / <& form> :=: <& form>),
              (not <* decl> = '%s, <& case> :=: 'basic / continue),
              nomen.avstavning,
              store, assign.majorprocess('np_noun);
)
```

c.

```
(define sve.gram-entry prop.case
 #u           <* morph.cat> = 'case.suff,
              <& case> :=: 'gen,
              nomen.avstavning,
              store, assign.majorprocess(ng_prop);
 #! morph.cat;
)
```

d.

```
(define sve.gram-entry store.single.morph
 #u           if <* char :whitespace> = '+
              then
                           ((<& word.cat> = 'noun / <& word.cat> = 'pnoun),
                           (nomen.avstavning, store // advance, store.single.morph)
                           / (<& word.cat> = 'verb / <& morph.cat> = 'verb.flex),
                           (verb.avstavning, store // advance, store.single.morph)
                           / (<& word.cat> = 'adj / <& morph.cat> = 'adj.flex),
                           (adj.avstavning, store // advance, store.single.morph)
                           / (<& word.cat> = 'adv / <& morph.cat> = 'adv.flex),
                           (adverb.avstavning, store // advance, store.single.morph)
                           / (store // advance, store.single.morph))
               else minorstore, process.dic;
)
```

Fig. 2.3: Rules changed to contain a call to the new rule nomen.avstavning.
a. close.noun.
b. store.noun.
c. prop.case.
d. store.single.morph, which also have a new calls to verb.avstavning, adj.avstavning and adverb.avstavning to differentiate between the hyphenation rules.

2.2.1 nomen.avstavning

The rule nomen.avstavning, (see Fig. 2.6), adds the attribute avstav, which is a concatenation of the stem and suffix-morphs with the separator, to the parse-structure.(Fig. 2.4.)

```
filmernas :
 (* = (1 = f
       2 = fi
       3 = fil
       4 = film
       LEM = FILM.NN
       DIC.STEM = <* 4>
       INFL = PATTERN.FILM
       STEM = <* 4>
       WORD.CAT = NOUN
       GENDER = UTR
       DECL = -ER
       FINAL = CONS
       NUMB = PLUR
       PLURMORF = er
       FORM = DEF
       FORMMORF = na
       CASE = GEN
       AVSTAV = film_er_nas))
```

Fig. 2.4: Complete parse-structure for the word ‘filmernas’ (the films´).

When there is no possible morphological hyphenation, the avstav attribute is omitted.
The rule nomen.avstavning also takes care of the instances where one or more morphs are missing or is not complete. i.e. words in plural only or when part of the morph has been consumed in the pattern rule. This is done by either assigning a minus-sign value or the wanted morph to the attributes plurmorf and/or formmorf. (See Fig. 2.5.)

a.

```
(define sve.gram-entry pattern.soli
 #u           assign.infl('soli),
              assign.dicstem,
              assign.stem,
              <& plurmorf> :=: '-,
              <& form> :=: 'indef,
              <& numb> :=: 'plur,
              noun('-,'neutr,'vowel);
)
```

b.

```
(define sve.gram-entry pattern.bord    ;;Hg 401-403,426;;
;; bord
 #u           assign.infl('bord),
              assign.dicstem,
              <& last.char> :=: <* char>,
              assign.stem,
              ((<& numb> :=: 'sing,
              noun('-,'neutr,'cons)
              // <& numb> :=: 'plur,
              (<& form> :=: 'indef
              // <& form> :=: 'def,
              <& formmorf> :=: '%e%n%a,
              <& plurmorf> :=: '-)),
              noun('-en,'neutr,'cons)
              // assign.compstem,
              (nolink
              // advance, <* char> = '%s,
              not <& last.char> = '%s, link(<* char>)));
)
```

Fig. 2.5: Assigning different values to the attributes plurmorf and formmorf;
a. Plural only, assigning minus to plurmorf.
b. Definite form plural, with single letter plural morph or form morph, assigning minus to plurmorf and assigning ‘ena’ to formmorf.

```
(define sve.gram-entry nomen.avstavning
 #u           (<& case> = 'gen,
              ((<& numb> = 'plur,
              <& form> = 'def,
              not <& formmorf> = '-,
              not <& plurmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& plurmorf> '_ <& formmorf> '%s /
              <& avstav> :=: <& stem> '_ <& plurmorf> '_ <& formmorf> '%s) /
              <& numb> = 'plur,
              <& form> = 'indef,
              not <& plurmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& plurmorf> '%s /
              <& avstav> :=: <& stem> '_ <& plurmorf> '%s))
              /
              <& form> = 'def,
              not <& formmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& formmorf> '%s /
              <& avstav> :=: <& stem> '_ <& formmorf> '%s))
              /
              <& numb> = 'plur,
              <& form> = 'def,
              not <& formmorf> = '-,
              (not <& plurmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& plurmorf> '_ <& formmorf> /
              <& avstav> :=: <& stem> '_ <& plurmorf> '_ <& formmorf>)
              / (<& astem>, <& avstav> :=: <& astem> '_ <& formmorf> /
              <& avstav> :=: <& stem> '_ <& formmorf>))
              /
              <& numb> = 'plur,
              <& form> = 'indef,
              (not <& plurmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& plurmorf> /
              <& avstav> :=: <& stem> '_ <& plurmorf>)
              / (<& astem>, <& avstav> :=: <& astem> /
              <& avstav> :=: <& stem>))
              /
              <& numb> = 'sing,
              <& form> = 'def,
              not <& formmorf> = '-,
              (<& astem>, <& avstav> :=: <& astem> '_ <& formmorf> /
              <& avstav> :=: <& stem> '_ <& formmorf>)
              /
              (<& astem>,
              (<& case> = 'gen, <& avstav> :=: <& astem> '%s /
              <& avstav> :=: <& astem>)
              / continue));
)
```

Fig. 2.6: The rule nomen.avstavning, which adds the avstav attribute to the parse-structure.

2.3 Changes to Noun Stems

In order to get the best morphological hyphenation, some stems were changed.

Old stem	**Example**	**New stem**	**Example**
bränder	bränder	bränd	bränd-er
böcker	böcker	böck	böck-er
bönder	bönder	bönd	bönd-er
centr	centr-um	cent	cent-rum
eko	ekot	ek	ek-ot
fäder	fäder	fäd	fäd-er
fängelse	fängelser	fängels	fängels-er
fötter	fötter	fött	fött-er
kölvatten	kölvatten	kölvatt	kölvatt-en
länder	länder	länd	länd-er
minim	minima	mini	mini-ma
mödrar	mödrar	möd*	möd-rar
nätter	nätter	nätt	nätt-er

område	områden	områd	områd-en
rötter	rötter	rött	rött-er
stränder	stränder	sträng	sträng-er
söner	söner	sön	sön-er
tänder	tänder	täng	täng-er
änder	änder	äng	äng-er

* Not yet implemented.

2.4 Stem-internal Hyphenation

The stem-internal hyphenation of nouns is handled by checking for the attribute astem, in nomen.avstavning, if present it is used in the concatenation instead of stem when assigning avstav. The attribute astem can be assigned either in the stem dictionary or in the inflection pattern-rules. (See Fig. 2.7.)

```
a.
(define sve.dic-entry "fjolåret" #u <& lem>:=:'fjolåret.nn, <& astem> :=: '%f%j%o%l_%å%r%e%t,
pattern.fjolåret;)

b.
(define sve.gram-entry pattern.segel           ;;Hg 431;;
 #u           assign.infl('segel),
              assign.dicstem,
              advance,
              (<* char> = '%e,
              advance, <* char> = '%l,
              <& astem> :=: <& dic.stem> '_%e%l,
              ...
)
```

Fig. 2.7: Two ways to assign the attribute astem for stem-internal hyphenation;
a. In the stem dictionary for the word ‘fjolåret’ (the last year)
b. In a pattern rule for the pattern ‘segel’ (sail). (Here partial.)

2.5 Noun Compounds

Noun compounds consisting of noun plus noun are treated in almost the same way as ordinary nouns, but the rule to handle the concatenation of the morphs is called comp.avstavning (compound hyphenation), which is called in the rule compound. (See Fig. 2.8.)

```
(define sve.gram-entry comp_noun
 #u           not <* compound>,
              <& compound>:=:'+,
              <* morph.cat> = 'comp.stem,
              <& first> :=:<*>,
              advance,
              <* word.cat> = 'noun,
              <&  word.cat> :=: 'noun,
              <& infl> :=: <* infl>,
              <& dic.stem> :=: <& first dic.stem> '+ <* dic.stem>,
              <& stem> :=: <& first stem><* dic.stem>,
              <& gender> :=: <* gender>,
              <& form> :=: <* form>,
              <& numb> :=: <* numb>,
              <& case> :=: <* case>,
              <& lem> :=: <& first lem> '+ <* lem>,
              comp.avstavning,
              assign.comp(abbrev),
              assign.comp(proper),
              assign.comp(prefix),
              store;

 #! morph.cat or word.cat;
)
```

Fig. 2.8: The rule comp_noun modified to contain a call to the new rule comp.avstavning.

2.5.1 comp.avstavning

The rule comp.avstavning, Fig. 2.9, concatenates the morphs of the noun compound:

```
(define sve.gram-entry comp.avstavning
 #u           (<& first astem>, <& förled> :=: <& first astem> /
              <& förled> :=: <& first stem>),
              (<* astem>, <& efterled> :=: <* astem> /
              <& efterled> :=: <* stem>),
              ((<& case> = 'gen,
              not <* formmorf> = '-,
              not <* plurmorf> = '-,
              (<& numb> = 'plur,
              <& form> = 'def,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* plurmorf> '_ <* formmorf> '%s /

              <& numb> = 'plur,
              <& form> = 'indef,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* plurmorf> '%s /

              <& numb> = 'sing,
              <& form> = 'def,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* formmorf> '%s /

              <& numb> = 'sing,
              <& form> = 'indef,
              <& avstav> :=: <& förled> '_ <& efterled> '%s)) /

              <& numb> = 'plur,
              <& form> = 'def,
              (not <* formmorf> = '-,
              (not <* plurmorf> = '-,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* plurmorf> '_ <* formmorf>
              / <& avstav> :=: <& förled> '_ <& efterled> '_ <* formmorf>) /
              <& avstav> :=: <& förled> '_ <& efterled>) /

              <& numb> = 'plur,
              <& form> = 'indef,
              (not <* plurmorf> = '-,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* plurmorf>
              / <& avstav> :=: <& förled> '_ <& efterled>)
              /
              <& numb> = 'sing,
              <& form> = 'def,
              not <* formmorf> = '-,
              <& avstav> :=: <& förled> '_ <& efterled> '_ <* formmorf> /

              <& numb> = 'sing,
              (<& form> = 'indef,
              not <& förled>,
              not <& efterled>
              / (<* formmorf> = '- / (<& förled> / <& efterled>)),
              (<& case> = 'basic,
              <& avstav> :=: <& förled> '_ <& efterled>
              / <& avstav> :=: <& förled> '_ <& efterled> '%s)));
)
```

Fig. 2.9: The new rule comp.avstavning, which hyphenates noun compounds.

The only difference between comp.avstavning and nomen.avstavning is that comp.avstavning always adds the attribute avstav to the parse-structure, since every compound can be hyphenated between the two stems. (See Fig. 2.10.)

```
filmnötterna :
 (* = (COMPOUND = +
       FIRST = (1 = f
                2 = fi
                3 = fil
                4 = film
                LEM = FILM.NN
                DIC.STEM = <* FIRST 4>
                INFL = PATTERN.FILM
                MORPH.CAT = COMP.STEM
                STEM = <* FIRST 4>)
       INFL = PATTERN.NÖT
       DIC.STEM = film+nöt
       STEM = filmnöt
       GENDER = UTR
       FORM = DEF
       NUMB = PLUR
       CASE = BASIC
       LEM = FILM.NN+NÖT.NN
       AVSTAV = film_nött_er_na))
```

Fig. 2.10: Parse-structure for the word ‘filmnötterna’ (the film nuts).

3. Verbs

The addition of morphological hyphenation rules for verbs will be outlined in this chapter. The implementation of the verb rules takes a slightly different approach than that of the noun rules, where the pattern rules were heavily changed.This different approach have leveled the changes to affect the morph dictionary-rules, pattern-rules and core verb-rules as little as possible more evenly.

3.1 Preparing Hyphenation/Syllabification of Verbs

In order to get at the suffixes, for hyphenation, the procedure is the same as for the nouns. The letters have to be saved during the dictionary consultation. The only new dictionary whose gathering-rule has to be changed is *verb-flex*, since the stem dictionary is the same for all word classes by default. In Fig. 3.1, there are two attributes pastpartmorf and adjmorf, which are assigned the value of morphs.
The inflmorf attribute is also available as in the noun rules. Here inflmorf is assigned either in the morph dictionary rules, or in the inflectional pattern rules, (Fig. 3.1b1-2), or in the rule verb.avstavning. (Fig. 3.4.)

a.

```
(define sve.gram-entry past.participle
 #u           <& numb>:=:<* numb>,
              <& def>:=:<* def>,
              <& gender>:=:<* gender>,
              <& func>:=:<* func>,
              assign(a-form),
              assign(sex),
              assign(case),
              ((<& pastpartmorf> / <& pastpartmorf> :=: <* :last>),
              <* eow>, store
              / advance, <* morph.cat>='adj.flex,
              <& conj>:=:<* conj>,
              assign(func),assign(numb),
              (<& conj> = 'I, assign(def),
              (<* eow>,<& case>:='basic, assign(a-form)
              / <& def>:='def, <& func>:='attr)
              / (<& conj> = 'II / <& conj> = 'III / <& conj> = 'IV),
              <& adjmorf> :=: <* :last>,
              (<* sex>, <& def> :=: 'def,
              <& gender> :=: 'utr, assign(sex)
              / assign(def), assign(gender),
              assign(func), assign(a-form))),
              (<* eow>,<& case>:='basic,
              store
              / advance,
              <* morph.cat> = 'case.suff,
              <& case> :=: 'gen,
              store));
)
```

b1.

```
(define verb.flex-entry "ande"
 #u           (<& conj> :=: 'II // <& conj> :=: 'IV),
              <& inff> :=: 'ap,
              <& morph.cat> :=: 'verb.flex,
              advance,
              (end.of.word,<& depon> :=: '-,
              <& case> :=: 'basic,
              <& inflmorf> :=: '%a%n_%d%e,
              store.final.morph
              / <* char> = '%s,
              advance, end.of.word,
              (<& depon> :=: '-,
              <& case> :=: 'gen
              / <& depon> :=: '+),
              <& inflmorf> :=: '%a%n_%d%e%s,
              store.final.morph);
)
```

b2.

```
(define sve.gram-entry pattern.brottas    ;;Hg 751-754;;
;;brott-a+s, brott-a+de+s, brott-a+t+s, brott-a+ndes
 #u           assign.infl('brottas),
              assign.dicstem,
              advance, <* char> = '%a,
              <& conj> :=: 'I, <& depon> :=: '+,
              advance,
              (<* char> = '%s,
              advance,end.of.word,
              (<& inff> :=: 'fin,<& tense> :=: 'pres
              // <& inff> :=: 'inf // <& inff> :=: 'imp),
              assign.majorprocess('vp),
              <& word.cat> :=: 'verb,
              <& diat> :=: 'depon,
              <& inflmorf> :=: '%a%s,
              store.single.morph
              / <& linkb> :=: '%a,
              (<* char> = '%n, <& inff>:=:'ap, verb
              / <* char> = '%d, <& inff>:=:'fin, verb
              / <* char> = '%t, <& inff>:=:'sup, verb));
)
```

Fig. 3.1: **a.** The rule past.participle. Changes (in italics) to save the suffix-morphs pastpartmorf and adjmorf, which also shows the interaction with the adjective rules. **b1-2.** Verb rules with the attribute inflmorf added. (Verb morph dictionary rule and pattern rule.)

3.1.1 The Attribute same

The attribute same and the rule assign.same is one of the changes in approach for the verbs. The attribute is added from the morph dictionary-rules, or the pattern-rules for forms which are not to be hyphenated. The rule assign.same, which checks for the attributes linkb, pastpartmorf or inflmorf. If any of these attributes is found, the assignment of a plus sign to same, to indicate that there should be no morphological hyphenation, is skipped since there are possible morphs to hyphenate in between. (See Fig. 3.2a.)

a.
```
(define sve.gram-entry assign.same
 #u           if <& linkb> / <* pastpartmorf> / <& inflmorf>
              then continue
              else <& same> :=: '+;
)
```

b.
```
(define verb.flex-entry "r"
 #u           advance, end.of.word,
              <& depon> :=: '-,
              <& morph.cat> :=: 'verb.flex,
              (<& conj> :=: 'I / <& conj> :=: 'III),
              <& inff> :=: 'fin,
              <& tense> :=: 'pres,
              <& diat> :=: 'act,
              assign.same,
              store.final.morph;
)
```

c.
```
(define sve.gram-entry pattern.älska    ;;Hg 711-720 ;;
;;älsk-a+r
 #u           assign.infl('älska),
              assign.dicstem,
              advance, <* char> = '%a,
              <& conj>:=:'I,
              <& depon>:=:'-,
              advance,
              (end.of.word,
              <& word.cat> :=:'verb,
              <& diat>:=:'act,
              (<& inff>:=:'inf
              // <& inff>:=:'imp),
              assign.majorprocess('vp),
              assign.same,
              store.single.morph
              / <& voice> :=: '+,
              <& linkb> :=: '%a,
              verb);
)
```

Fig. 3.2: The new rule assign.same, (**a**.), and examples of where it is called from (**b-c**.). In **c.** there is an example of the new attribute linkb too.

3.1.2 The Attribute linkb

linkb is an attribute for saving letters which otherwise would be lost in checks in the pattern rules. (See Fig. 3.2c.) The value of linkb is concatenated to the morph in verb.avstavning. (Fig. 3.4.) It also blocks the assignment of same, as mentioned in the previous section.

3.2 Putting the Verb-morphs together

The verb rule inflected.verb, needed some additional changes to be able to handle the hyphenation, since some attributes were not available when verb.avstavning was to be called. The attributes are as seen in Fig. 3.3; stem, inflmorf, linkb and same.
To cover all ways to end parsing of verbs, a call to the new rule verb.avstavning (verb hyphenation) were added to the rules in Fig. 3.3 and 2.3d. (store.single.morph):

```
(define sve.gram-entry inflected.verb
 #u           <* morph.cat> = 'verb.stem,
              <& lem> :=: <* lem>,
              <& dic.stem> :=: <* dic.stem>,
              (<* same> = '+, <& same> :=: <* same> / continue),
              (<* stem>, <& stem> :=: <* stem> / continue),
              (<* inflmorf>, <& inflmorf> :=: <* inflmorf> / continue),
              (<* linkb>, <& linkb> :=: <* linkb> / continue),
              <& conj> :=: <* conj>,
              <& infl> :=: <* infl>,
              <& depon> :=: <* depon>,
              <& inff> :=: <* inff>,
              <& diat> :=: <* diat>,
              advance,
              <* morph.cat> = 'verb.flex,
              <& conj> :=: <* conj>,
              <& depon> :=: <* depon>,
              (<* same> = '+, <& same> :=: <* same> / continue),
              (<* inflmorf>, <& inflmorf> :=: <* inflmorf> / continue),
              (<* depon> = '+, <& diat> :=: 'depon / continue),
              <& inff> :=: <* inff>,
              <& word.cat> :=: 'verb,
              (<& inff> = 'inf, <& diat>:=:<* diat>, store,
              assign.majorprocess('vp_part)
              / <& inff> = 'fin, <& diat>:=:<* diat>, finite.verb
              / <& inff> = 'sup, <& diat>:=:<* diat>, supine
              / <& inff> = 'pp, assign.majorprocess('vp_part), past.participle
              / <& inff> = 'ap, <& case> :=: <* case>,
              store, assign.majorprocess('vp_part)
              / <& inff> = 'conj, store),
              verb.avstavning;

 #! morph.cat;
)
```

Fig. 3.3: Rule that has got a call to verb.avstavning added to it;
verb.inflection. Notice the other changes made (in italics).

As seen in Fig. 2.3d store.single.morph has to check the word class of the word, in order to decide which hyphenation rule to call. Some verb inflection-patterns did not assign the value verb to the attribute word.cat, so a check for the value verb.flex of the attribute morph.cat has been added to compensate for this.

3.2.1 verb.avstavning

Now with the verb-morphs available in the attributes dic.stem (or stem), inflmorf, pastpartmorf, adjmorf and linkb and the attribute same not being positive (a plus-sign), they have to be put together with the separator in between them. This is done in verb.avstavning, (Fig. 3.4.), where the attribute inflmorf is assigned the value of :last if it does not already have a value.

```
(define sve.gram-entry verb.avstavning
 #u           (<& inflmorf>
              / not <* :last> = 'nil, <& inflmorf> :=: <* :last>
              / continue),
              (<& pastpartmorf>,
              (<& adjmorf>,
              (<& case> = 'gen,
              <& suffix> :=: <& pastpartmorf> <& adjmorf> '%s
              / <& suffix> :=: <& pastpartmorf> <& adjmorf>)
              / (not <& inflmorf> = 'nil,
              not <& pastpartmorf> :=: <& inflmorf>,
              <& suffix> :=: <& pastpartmorf> <& inflmorf>
              / <& suffix> :=: <& pastpartmorf>))
              / continue),
              (not <& suffix>, <& suffix> :=: <& inflmorf> / continue),
              (not <& suffix> = 'nil,
              not <& same> = '+,
              (not <& linkb> = 'nil,
              (<& stem>, <& avstav> :=: <& stem> '_<& linkb><& suffix>
              / <& avstav> :=: <& dic.stem> '_ <& linkb><& suffix>)
              / (<& stem>, <& avstav> :=: <& stem> '_<& suffix>
              / <& avstav> :=: <& dic.stem> '_ <& suffix>))
              / continue);
)
```

Fig. 3.4: The new rule verb.avstavning.

If the value of same is not a plus sign, the attribute avstav is added to the parse output structure, containing the morphologically hyphenated form, see Fig. 3.5:

```
brottandes :
 (* = (LEM = BROTTAS.VB
       DIC.STEM = brott
       LINKB = a
       CONJ = I
       INFL = PATTERN.BROTTAS
       DEPON = +
       INFF = AP
       DIAT = DEPON
       INFLMORF = ndes
       WORD.CAT = VERB
       CASE = NIL
       SUFFIX = <* INFLMORF>
       AVSTAV = brott_andes))
```

Fig. 3.5: The parse structure for the word brottandes (‘being wrestling’).

3.3 Changes to Verb Stems

In order to get the best morphological hyphenation some stems were changed.

Old stem	**Example**	**New stem**	**Example**
älska	älska-de	älsk	älsk-ade
brotta	brotta-de	brott	brott-ade

4. Adjectives

The addition of morphological hyphenation rules for adjectives will be outlined in this chapter.
The implementation of the adjective rules follows the approach of the verb rules.

4.1 Preparing Hyphenation/Syllabification of Adjectives

In order to get at the suffixes, for hyphenation, the procedure is the same as for the other word classes. The letters have to be saved during the dictionary consultation. There are two dictionaries whose gathering-rule has to be changed; adj-*flex* and *adj.flex.e.*
In Fig. 4.1a, the attribute supmorf, is assigned the value of the supine suffix morph.
The inflmorf attribute is also available as in the noun rules. Here inflmorf is assigned either in the morph dictionary-rules, or in the inflectional pattern-rules (Fig. 4.1b1-3) or in the rule adj.avstavning. (Fig. 4.5.)

a.
```
(define sve.gram-entry store.adj.supinum
 #u           <& degree> :=: 'sup,
              advance,
              <& morph.cat> :=: 'adj.flex,
              (end.of.word, <& eow>:=:'*, <& func> :=: 'pred,
              store.adj
              / <& func> :=: 'attr,
              av.prop2,
              <& supmorf> :=: <& inflmorf>,
              minorstore,
              process(adj.flex.e,<prop>));
)
```

b1.
```
(define adj.flex-entry "are"
 #u           <& degree> :=: 'comp,
              <& comp.type> :=: 'are,
              <& degree> :=: 'comp,
              <& inflmorf> :=: '%a%r%e,
              store.adj.flex;
)
```

b2.
```
(define sve.gram-entry pattern.angelägen ;;Hg 613
;; angeläg-(e)-n,angeläg-e-t,angeläg-n+a
 #u           assign.infl('angelägen),
              assign.dicstem,
              advance,
              (<* char> = '%e, advance,
              (<* char> = '%t, <& inflmorf> :=: '%e%t, adj.sing.pos('neutr)
              / <* char> = '%n, <& inflmorf> :=: '%e%n, adj.sing.pos('utr))
              / <* char> = '%n, <& linkb> :=: '%n, adj('are));
)
```

b3.
```
(define sve.gram-entry pattern.mindre
;; mindre -> mind-re
 #u           assign.infl('mindre),
              assign.dicstem,
              <& degree> :=: 'comp,
              advance,
              (<* char> = '%a,
              advance, <* char> = '%r,
              advance, <* char> = '%e,
              <& inflmorf> :=: '%a%r%e
              / <* char> = '%d,
              advance, <* char> = '%r,
              advance, <* char> = '%e,
              <& stem> :=: <& dic.stem> '%d,
              <& inflmorf> :=: '%r%e),
              adj.form;
)
```

Fig. 4.1: **a.** The rule store.adj.supinum. Changed to save the supine suffix-morph in attribute supmorf.
b1-3. Adjective rules with the attribute inflmorf added. (One morph dictionary rule and two different pattern rules.)

4.1.1 The Attribute same

The attribute same and the rule assign.same is the same as for the verb rules. See section 3.1.1.

a.
```
(define adj.flex-entry "t"
 #u           <& stem.type> :=: 'cons,
              assign.same,
              store.adj.flex-t;
)
```

b.
```
(define sve.gram-entry pattern.ringa  ;;Hg 631;;
;; SOB: "adj., neutr. och plur. ="
;; ringa -> ringa, ringare, ringast, ringaste
 #u           assign.infl('ringa),
              assign.dicstem,
              advance, <* char> = '%a,
              <& word.cat> :=: 'adj,
              (<& degree> :=: 'pos,
              advance,
              assign.same,
              store.adj
              // (<&  degree> :=: 'comp // <&  degree> :=: 'sup),
              <& linkb> :=: '%a,
              adj('re));
)
```

c.
```
(define sve.gram-entry no.adj.flex
 #u           end.of.word,
              <& degree> :=: 'pos,
              <& numb> :=: 'sing,
              <& form> :=: 'indef,
              <& case> :=: 'basic,
              <& a-form> :=: 'nil,
              (<& func> / <& func> :=: 'nil),
              <& sex> :=: 'nil,
              (if not (<& stem.char>='t / <& stem.char>='tt)
                        then <& gender> :=: 'utr else continue),
              assign.same,
              store.adj;
)
```

Fig. 4.2: The use of the new attribute same with adjective rules;
a. morph dictionary rule

b. inflectional pattern rule
c. core adjective rule

4.1.2 The Attribute linkb

The use of linkb is the same as for the verb rules. (See Fig. 4.2b. and section 3.1.2.)

4.2 Putting the Adjective-morphs together

The adjective rules av.prop1, a.adjfeat and adj.flexion, needed some changes to be able to handle the new attributes for hyphenation, since some attributes were not available when adj.avstavning was to be called. The attributes, linkb and supmorf are seen in Fig. 4.3 and the attribute same is seen in Fig.4.4.

a.
```
(define sve.gram-entry av.prop1
 #u           <prop degree> :=: <& degree>,
              <prop comp.type> :=: <& comp.type>,
              <prop stem.type> :=: <& stem.type>,
              <prop stem.char> :=: <& stem.char>,
              <prop conj> :=: <& conj>,
              <prop linkb> :=: <& linkb>,
              ((<& stem.char> = 'd / <& stem.char> = 'tt),
              <prop -d.tt> :=: '+
              / <prop -d.tt> :=: '-),
              (<& pp>, <prop pp> :=: <& pp>
              / <prop pp> :=: '-),
              (<& dd>, <prop dd> :=: <& dd>
              / <prop dd> :=: '-);
)
```

b.
```
(define sve.gram-entry a.adjfeat
 #u     assign(degree),
              assign(gender),
              assign(numb),
              assign(form),
              assign(a-form),
              assign(case),
              assign(sex),
              assign(func),
              (<* supmorf>, <& supmorf> :=: <* supmorf> / continue);
)
```

Fig. 4.3: Rules that have been changed to save new attributes to allow hyphenation;
a. av.prop1. Saves linkb.
b. a.adjfeat. Saves supmorf in order to handle supine before adjectival endings.

To cover all ways to end parsing of verbs, calls to the new rule adj.avstavning (adjective hyphenation) was added to the rules in Fig. 4.4 and Fig. 2.3d (store.single.morph):

```
(define sve.gram-entry adj.flexion
 #u           <* morph.cat> = 'adj.flex,
              a.adjfeat,
              (<* eow>, close.adj,
              (<& a-form> = 't, not <& lem :adv> = '-,
              assign.majorprocess('advp) / continue),
              (<* same>, <& same> :=: <* same> / continue)
              / advance,
              (<* morph.cat> = 'adj.flex,
              a.adjfeat,
              (<* eow>, close.adj / advance, adj.case))),
              adj.avstavning;

 #!           morph.cat;
)
```

Fig. 4.4: The rule adj.flexion with the call to adj.avstavning and the unification of the attribute same added.

As seen in Fig. 2.3d store.single.morph has to check the word class of the word, in order to decide which hyphenation rule to call. Some adjectival inflection-patterns did not assign the value adj to the attribute word.cat, so a check for the value adj.flex of the attribute morph.cat has been added to compensate for this.

4.2.1 adj.avstavning

Now with the adjective-morphs available in the attributes dic.stem (or stem), inflmorf and linkb and the attribute same not being positive (a plus sign), they have to be put together with the separator in between them. This is done in adj.avstavning, (Fig. 4.5.), where the attribute inflmorf is assigned the value of :last if it does not already have a value.

```
(define sve.gram-entry adj.avstavning
 #u           (not <& same> = '+,
              (<& inflmorf>
              / not <* :last> = 'nil, <& inflmorf> :=: <* :last>
              / continue),
              (<& supmorf>,
              (<& inflmorf>,
              (<& case> = 'gen,
              <& suffix> :=: <& supmorf> <& inflmorf> '%s
              / <& suffix> :=: <& supmorf> <& inflmorf>))
              / continue),
              (not <& suffix>, <& suffix> :=: <& inflmorf> / continue),
              (not <& suffix> = 'nil,
              (not <& linkb> = 'nil,
              (<& stem>, <& avstav> :=: <& stem> '_<& linkb><& suffix>
              / <& avstav> :=: <& dic.stem> '_ <& linkb><& suffix>)
              / (<& stem>, <& avstav> :=: <& stem> '_<& suffix>
              / <& avstav> :=: <& dic.stem> '_ <& suffix>)))
              / continue);
)
```

Fig. 4.5: The new rule adj.avstavning.

If the value of same is not a plus sign, the attribute avstav is added to the parse structure, containing the morphologically hyphenated form, see Fig. 4.6:

```
lättare :
 (* = (1 = l
       2 = lä
       3 = lät
       4 = lätt
       LEM = LÄTT.AV
       INFL = PATTERN.LÄTT
       STEM.CHAR = TT
       GENDER = NIL
       DIC.STEM = <* 4>
       WORD.CAT = ADJ
       COMP.TYPE = ARE
       DEGREE = COMP
       NUMB = NIL
       FORM = NIL
       A-FORM = NIL
       CASE = BASIC
       SEX = NIL
       FUNC = NIL
       INFLMORF = are
       SUFFIX = <* INFLMORF>
       AVSTAV = lätt_are))
```

Fig. 4.6: The parse structure for lättare (‘easier’).

4.3 Changes to Adjective Stems

In order to get the best morphological hyphenation some stems were changed.

Old stem	**Example**	**New stem**	**Example**
ringa	ring	ring	ring-are
minst	minst	min	min-sta
mindre	mindre	min	min-dre
närmare	närmare	närm	närm-are
närmast	närmast	närm	närm-ast
mellersta	mellersta	meller	meller-sta
flest	flest	fles	fles-ta

5. Adverbs

The addition of morphological hyphenation rules for adverbs will be outlined in this chapter. The implementation of the adverb hyphenation rules follows the approach of the verb rules.

5.1 Preparing Hyphenation/Syllabification of Adverbs

In order to get at the suffixes, for hyphenation, the procedure is the same as for the nouns. The letters have to be saved during the dictionary consultation. The only new dictionary whose gathering-rule has to be changed is adv-*flex*, since the stem dictionary is the same for all word classes by default.
The inflmorf attribute is also available as in the noun rules. Here inflmorf is assigned either in the inflectional pattern-rules or in the rule adverb.avstavning.

a.
```
(define sve.gram-entry adverb.avstavning
 #u           (not <& same> = '+,
              (<& inflmorf>
              / not <* :last> = 'nil, <& inflmorf> :=: <* :last>
              / <& same> :=: '+, continue),
              (not <& inflmorf> = 'nil,
              (not <& linkb> = 'nil,
              (<& stem>, <& avstav> :=: <& stem> '_<& linkb><& inflmorf>
              / <& avstav> :=: <& dic.stem> '_ <& linkb><& inflmorf>)
              / (<& stem>, <& avstav> :=: <& stem> '_<& inflmorf>
              / <& avstav> :=: <& dic.stem> '_ <& inflmorf>)))
              / continue);
)
```

b.
```
(define sve.gram-entry pattern.vidare
;; vidare
 #u           assign.infl('vidare),
              assign.dicstem,
              advance, <* char> = '%a,
              advance,
              (<* char> = '%r,
              advance, <* char> = '%e,
              <& inflmorf> :=: '%a%r%e
              /<* char> = '%s,
              advance, <* char> = '%t,
              advance, <* char> = '%e,
              <& inflmorf> :=: '%a%s%t%e),
              <& word.cat>:=:'adv,
              <& degree>:=:'comp,
              <& comp.type>:=:'re,
              advance,end.of.word,
              store.single.morph;
)
```

Fig. 5.1: **a.** The new rule adverb.avstavning. The only place for the adverb rules to save suffix morphs.
b. An example of the use of the attribute inflmorf, in a pattern rule.

5.1.1 The Attribute same

The attribute same and the rule assign.same is the same as for the verb rules. (See section 3.1.1 and Fig. 5.2.)

a.
```
(define sve.gram-entry store.adv
 #u           assign.same,
              store.single.morph,
              if <& lex>,<& lex :quant>
         then assign.majorprocess('qp)
         else assign.majorprocess('advp);

 #! char;
)
```

b.
```
(define sve.gram-entry pattern.bättre.ab
;; bä-tt+re, bä+st
 #u           assign.infl('bättre.ab),
              (<& degree>:=:'sup,
              assign.same
              // advance, <* char> = '%t,
              advance, <* char> = '%t,
              <& degree> :=: 'comp),
              adverb('re);
)
```

Fig. 5.2: The rule assign.same and examples of where it is called from;
a. core adverb rule
b. inflectional pattern rule

5.1.2 The Attribute linkb

The attribute linkb is used the same way as in the verb rules. (See Fig. 5.3 and section 3.1.2) The value of linkb is concatenated to the morph in adverb.avstavning. (Fig. 5.1.)

```
(define sve.gram-entry pattern.ofta
;; ofta, ofta+re, ofta+st
 #u           assign.infl('ofta),
              advance, <* char> = '%a,
              <& linkb> :=: '%a,
        adverb('re);
)
```

Fig. 5.3: The use of linkb in an adverb inflectional pattern-rule.

5.2 Putting the Adverb-morphs together

The adverb rule adv.flexion calls the adverb.avstavning. (Fig. 5.4.)

```
(define sve.gram-entry adv.flexion
 #u           <* morph.cat>='adv.flex,
              <& degree>:=:<* comp>,
              store,
              assign.majorprocess(advp),
              adverb.avstavning;
 #! morph.cat;
)
```

Fig. 5.3: The rule adv.flexion that has got a call to adverb.avstavning added to it.

As seen in Fig. 2.3d, store.single.morph has to check the word class of the word, in order to decide which hyphenation rule to call. Some adverb inflection-patterns did not assign the value adv to the attribute word.cat, so a check for the value adv.flex of the attribute morph.cat has been added to compensate for this.

5.2.1 adverb.avstavning

Now with the verb-morphs available in the attributes dic.stem (or stem), linkb and inflmorf and the attribute same not being positive (a plus sign), they have to be put together with the separator in between them. This is done in adverb.avstavning, (Fig. 5.1), where the attribute inflmorf is assigned the value of :last if it does not already have a value.
If the value of same is not a plus sign, the attribute avstav is added to the parse structure, containing the morphologically hyphenated form, see Fig. 5.5:

```
fortare :
 (* = (1 = f
       2 = fo
       3 = for
       4 = fort
       LEM = FORT.AB
       INFL = PATTERN.FORT
       DIC.STEM = <* 4>
       WORD.CAT = ADV
       COMP.TYPE = ARE
       DEGREE = NIL
       INFLMORF = are
       AVSTAV = fort_are))
```

Fig. 5.5: The parse structure for the word fortare (‘faster’).

5.3 Changes to Adverb Stems

In order to get the best morphological hyphenation some stems were changed.

Old stem	Example	New stem	Example
ofta	ofta	oft	oft-ast
vidare	vidare	vid	vid-are
mitterst	mitterst	mitt	mitt-erst

6. A few words about the future

Add hyphenation to compound rules for other word classes and combinations of word classes than noun plus noun when this is implemented. Implement the mechanical hyphenation as a complement to the output of these morphological rules.

References

[1] "Svenska skrivregler", utgivna av Svenska språknämnden, Almqvist & Wiksell, 1991

[2] "Svensk avstavningsordbok", R. Klingspor, Tryckeriförlaget, 1989, ISBN 91-0081-6-8

[3] "Parsing by means of Uppsala Chart Processor (UCP)", A. Sågvall-Hein, Dept. of Linguistics, Uppsala, 1990

[WED98] Error Corpora Database, Wedbjer Rambell et al, http://www.stp.ling.uu.se/~olgaw/errortyp/index.html

Appendix A: Test Nouns and their Hyphenations

This is the output of the final test-run, July 22:nd, of the file noun.test through the avstav function. The test word comes first, then the morphologically hyphenated version or if this is not possible the test word again. The proper nouns tested produced a nil result, since they do not have a stem attribute and should, if possible, not be hyphenated.

ros	flick_or	tofflas	stol	dagens
ros	flickorna	toffla	stolens	dag_ens
rosen	flick_or_na	tofflans	stol_ens	dans
ros_en	flickas	toffl_ans	stolarnas	dan
rosor	flicka	tofflors	stol_ar_nas	dagars
ros_or	flickans	toffl_ors	stolars	dag_ars
rosorna	flick_ans	tofflornas	stol_ars	dagarnas
ros_or_na	flickors	toffl_or_nas	dag	dag_ar_nas
rosens	flick_ors	åthävor	dag	öken
ros_ens	flickornas	åthäv_or	dagen	ök_en
rosornas	flick_or_nas	stol	dag_en	öknen
ros_or_nas	toffla	stol	dan	ök_nen
rosors	toffla	stolen	dan	öknar
ros_ors	tofflor	stol_en	dagar	ök_nar
flicka	toffl_or	stolar	dag_ar	öknarna
flicka	tofflorna	stol_ar	dagarna	ök_nar_na
flickan	toffl_or_na	stolarna	dag_ar_na	öknens
flick_an	tofflan	stol_ar_na	dags	ök_nens
flickor	toffl_an	stols	dag	öknarnas

ök_nar_nas
ökens
ök_ens
öknars
ök_nars
afton
afton
aftonen
afton_en
aftnar
aft_nar
aftnarna
aft_nar_na
aftons
afton
aftonens
afton_ens
aftnars
aft_nars
aftnarnas
aft_nar_nas
morgon
morgon
morgonen
morgon_en
morgnar
morg_nar
morgnarna
morg_nar_na
mornar

morn_ar
mornarna
morn_ar_na
morgons
morgon
morgonens
morgon_ens
morgnarnas
morg_nar_nas
morgnars
morg_nars
mornars
morn_ars
mornarnas
morn_ar_nas
djävul
djävul
djävulen
djävul_en
djävlar
djäv_lar
djävlarna
djäv_lar_na
djävuls
djävul
djävulens
djävul_ens
djävlars
djäv_lars
djävlarnas
djäv_lar_nas
kam
kam
kammen
kam_men
kammar
kam_mar
kammarna
kam_mar_na
kams
kam
kammens
kam_mens
kammars
kam_mars
kammarnas
kam_mar_nas
fru
fru
frun
frun
fruar
fru_ar
fruarna
fru_ar_na
frus
fru
fruns
frun
fruars
fru_ars
fruarnas
fru_ar_nas
gosse
gosse
gossen
goss_en
gossar
goss_ar
gossarna
goss_ar_na
gosses
gosse
gossens
goss_ens
gossars
goss_ars
gossarnas
goss_ar_nas
nyckel
nyckel
nyckeln
nyck_eln
nycklar
nyck_lar
nycklarna
nyck_lar_na
nyckels
nyckel
nyckelns
nyck_elns
nycklars
nyck_lars
nycklarnas
nyck_lar_nas
seger
seg_er
segern
seg_ern
segrar
seg_rar
segrarna
seg_rar_na
segers
seg_ers
segerns
seg_erns
segrars
seg_rars
segrarnas
seg_rar_nas
lämmel
lämmel
lämlar
läm_lar
lämmeln
läm_meln
lämlarna
läm_lar_na
lämmels
lämmel
lämmelns
läm_melns
lämlars
läm_lars
lämlarnas
läm_lar_nas
hummer
hummer
hummern
hum_mern
humrar
hum_rar
humrarna
hum_rar_na
hummerns
hum_merns
hummers
hummer
humrars
hum_rars
humrarnas
hum_rar_nas
finger
fing_er
finger
fing_er
fingret
fing_ret
fingrar
fing_rar
fingrarna
fing_rar_na
fingers

fing_ers
fingers

fing_ers
fingrars
fing_rars
fingrarnas
fing_rar_nas
fingern

fing_ern
fingerns
fing_erns
fingrets
fing_rets
botten
botten
bottnen
bott_nen
bottnar
bott_nar
bottnarna
bott_nar_na
bottens
botten
bottnens
bott_nens
bottnarnas
bott_nar_nas
bottnars
bott_nars
fröken
fröken
fröknar
frök_nar
fröknarna
frök_nar_na
frökens
fröken
fröknars
frök_nars
fröknarnas
frök_nar_nas
sommar
sommar
sommarn
sommarn
sommaren
sommar_en
somrar
som_rar
somrarna
som_rar_na
sommars
sommar
sommarns
sommarn
sommarens
sommar_ens
somrars
som_rars
somrarnas
som_rar_nas
himmel
himmel
himmelen
himmel_en
himmeln
himmeln
himlen
him_len
himlar
him_lar
himlarna
him_lar_na
himlens
him_lens
himlarnas
him_lar_nas
himmels
himmel
himmelens
himmel_ens
himmelns
himmeln
himlars
him_lars
tok
tok
token
tok_en
tokar
tok_ar
tokarna
tok_ar_na
toker
toker
tokern
tok_ern
toks
tok
tokers
toker
tokens
tok_ens
tokerns

tok_erns
tokars
tok_ars
tokarnas
tok_ar_nas
moder
moder
modern
modern
mödrar
mödr_ar
mödrarna
mödr_ar_na
moders
moder
moderns
moderns
mödrars
mödr_ars
mödrarnas
mödr_ar_nas
vägnar
vägn_ar
film
film
filmen
film_en
filmer
film_er
filmerna
film_er_na
films
film
filmens
film_ens
filmers
film_ers
filmernas
film_er_nas
vän
vän
vännen
vänn_en
vänner
vänn_er
vännerna
vänn_er_na
väns
vän
vännens
vänn_ens
vänners
vänn_ers
vännernas
vänn_er_nas
nöt
nöt
nöten
nöt_en
nötter
nött_er
nötterna
nött_er_na
nöts
nöt
nötens
nöt_ens
nötters
nött_ers
nötternas
nött_er_nas
vin
vin
vinet
vin_et
viner
vin_er
vinerna
vin_er_na
vins
vin
vinets
vin_ets
viners
vin_ers
vinernas
vin_er_nas
decennium
decenni_um
decennier
decenni_er
decennierna
decenni_er_na
decenniums
decenni_ums
decenniers
decenni_ers
decenniernas
decenni_er_na
s
decenniumet
decenni_um_et
decenniumets
decenni_um_et
s
geranium
gerani_um
geranien
gerani_en
geranier
gerani_er
geranierna
gerani_er_na
geraniums
gerani_ums
geraniers
gerani_ers
geraniens
gerani_ens
geraniernas
gerani_er_nas
aktivum
aktiv_um
aktivet
aktiv_et
aktiver
aktiv_er
aktiverna
aktiv_er_na
aktivums
aktiv_ums
aktivets
aktiv_ets
aktivers
aktiv_ers
aktivernas
aktiv_er_nas
futurum
futur_um
futurer
futur_er
futurerna
futur_er_na
futurums
futur_ums
futurers
futur_ers
futurernas
futur_er_nas
remedium
remedium
remediumet
remedium_et
remediet
remedi_et
remedier
remedi_er
remedierna
remedi_er_na
remediums
remediums
remediumets
remedium_ets
remediets
remedi_ets
remediers
remedi_ers
remediernas
remedi_er_nas
nivå
nivå
nivån
nivån
nivåer
nivå_er
nivåerna
nivå_er_na
nivås
nivå
nivåns
nivån
nivåers
nivå_ers
nivåernas
nivå_er_nas
akademi
akademi
akademien
akademi_en
akademin
akademi_n
akademin
akademin
akademier
akademi_er
akademier
akademi_er
akademierna
akademi_er_na
akademierna
akademi_er_na
akademis
akademi
akademiens
akademi_ens
akademins
akademin
akademins
akademi_ns
akademiers
akademi_ers
akademiers
akademi_ers
akademiernas
akademi_er_na
s
akademiernas
akademi_er_na
s
historia
historia
historian
historian
historien
histori_en
historier
histori_er
historierna
histori_er_na
historias
historia
historians
historian
historiens
histori_ens
historiers
histori_ers
historiernas
histori_er_nas
parti
parti
partiet
parti_et
partit
partit
partier
parti_er
partierna
parti_er_na
partis
parti
partiets
parti_ets
partits
partit
partiers
parti_ers
partiernas
parti_er_nas
möbel
möbel
möbeln
möbeln
möbler
möbl_er
möblerna
möbl_er_na
möbels
möbel
möbelns
möbeln
möblers
möbl_ers
möblernas
möbl_er_nas
musa
musa
musan
musan
muser
mus_er
muserna
mus_er_na
musas
musa
musans
musan
musers
mus_ers
musernas
mus_er_nas
drama
drama
dramat
dram_at

dramer
dram_er
dramerna
dram_er_na
dramas
drama
dramats
dram_ats
dramers
dram_ers
dramernas
dram_er_nas
ganglion
gangli_on
gangliet
gangli_et
ganglier
gangli_er
ganglierna
gangli_er_na
ganglions
gangli_ons
gangliets
gangli_ets
gangliers
gangli_ers
gangliernas
gangli_er_nas
strand
strand
stranden
strand_en
stränder
sträng_er
stränderna
sträng_er_na
strands
strand
strandens
strand_ens
stränders
sträng_ers
strändernas
sträng_er_nas
stad
stad
staden
stad_en
stan
stan
städer
städ_er
städerna
städ_er_na
stads
stad
stadens
stad_ens
stans
stan
städers
städ_ers
städernas
städ_er_nas
land
land
landet
land_et
länder
länd_er
länderna
länd_er_na
lands
land
landets
land_ets
länders
länd_ers
ländernas
länd_er_nas
kläder
kläd_er
kläderna
kläd_er_na
kläders
kläd_ers
klädernas
kläd_er_nas
sko
sko
skon
skon
skor
skor
skorna
skor_na
skos
sko
skons
skon
skors
skor
skornas
skor_nas
fängelse
fängelse
fängelset
fängels_et
fängelser
fängels_er
fängelserna
fängels_er_na
fängelses
fängelse
fängelsets
fängels_ets
fängelsers
fängels_ers
fängelsernas
fängels_er_nas
bonde
bonde
bonden
bonden
bönder
bönd_er
bönderna
bönd_er_na
bondes
bonde
bondens
bondens
bönders
bönd_ers
böndernas
bönd_er_nas
samhälle
samhälle
samhället
samhäll_et
samhällen
samhäll_en
samhällena
samhäll_ena
samhälles
samhälle
samhällets
samhäll_ets
samhällens
samhäll_ens
samhällenas
samhäll_enas
bi
bi
biet
bi_et
bin
bin
bina
bi_na
bis
bi
biets
bi_ets
bins
bin
binas
bi_nas
knä
knä
knät
knät
knän
knän
knäna
knä_na
knäs
knä
knäts
knät
knäns
knän
knänas
knä_nas
styck
styck
stycket
styck_et
stycken
styck_en
styckena
styck_ena
stycks
styck
styckets
styck_ets
styckens
styck_ens
styckenas
styck_enas
bord
bord
bordet
bord_et
borden
bord_en
bords
bord
bordets
bord_ets
bordens
bord_ens
vakuum
vakuum
vakuum
vakuum
vakuumet
vakuum_et
vakuumen
vakuum_en
vakuums
vakuum
vakuums
vakuum
vakuumets
vakuum_ets
vakuumens
vakuum_ens
gisslan
gisslan
gisslans
gisslan
order
order
ordern
ordern
orderna
ordern
orders
order
orderns
ordern
ordernas
ordern
fader
fader
far
far
fadern
fadern
fäder
fäd_er
fäderna
fäd_er_na
faders
fader
fars
far
faderns
faderns
fäders
fäd_ers
fädernas
fäd_er_nas
ordförande
ordförande
ordföranden
ordföranden
ordförandena
ordföranden
ordförandes
ordförande
ordförandens
ordföranden
ordförandenas
ordföranden
LP
LPn
LP
LP
LP-n
LP-n
LP-na
LP-_na
LPs
LP
LPs
LPn
LP-ns
LP-n
LP-nas
LP-_nas
pansar
pansar
pansar

pansar
pansaret
pansar_et
pansarna
pansar_na
pansars
pansar
pansars
pansar
pansarets
pansar_ets
pansarnas
pansar_nas
aber
aber
garage
garage
garaget
garaget
garagen
garagen
garages
garage
garagets
garaget
garagens
garagen
mil
mil
milen
mil_en
milen
mil_en
mils
mil
milens
mil_ens
milens
mil_ens
kypare
kypar
kyparen
kypar_en
kyparen
kypar_en
kyparna
kypar_na
kypares
kypar
kyparens
kypar_ens
kyparens
kypar_ens
kyparnas
kypar_nas
tecken
teck_en
tecknet
teck_net

tecknen
teck_nen
teckens
teck_ens
tecknets
teck_nets
tecknens
teck_nens
program
program
programmet
program_met
programmen
program_men
programs
program
programmets
program_mets
programmens
program_mens
tum
tum
tummen
tum_men
tummen
tum_men
tums
tum
tummens
tum_mens
tummens
tum_mens
segel
seg_el
seglet
seg_let
seglen
seg_len
segels
seg_els
segels
seg_el
seglets
seg_lets
seglens
seg_lens
fönster
fönst_er
fönster
fönst_er
fönstret
fönst_ret
fönstren
fönst_ren
fönsters
fönst_er
fönsters
fönst_ers
fönstrets

fönst_rets
fönstrens
fönst_rens
nummer
num_mer
numret
num_ret
numren
num_ren
nummers
num_mers
numrets
num_rets
numrens
num_rens
öga
öga
ögat
ög_at
ögon
ög_on
ögonen
ög_on_en
ögas
öga
ögats
ög_ats
ögons
ög_on
ögonens
ög_on_ens
gås
gås
gåsen
gås_en
gäss
gäss
gässen
gäss_en
gåsens
gås_ens
gässens
gäss_ens
man
man
mannen
mann_en
män
män
männen
männ_en
mans
mans
mannens
mann_ens
mäns
män
männens
männ_ens

ordalag
ordalag
ordalag
ordalag
ordalagen
ordalag_en
ordalagen
ordalag_en
jeans
jeans
jeansen
jeans_en
narkotikum
narkotik_um
narkotikumet
narkotik_um_e
t
narkotika
narkotika
narkotika
narkotika
narkotikan
narkotik_an
narkotikums
narkotik_ums
narkotikumets
narkotik_um_e
ts
narkotikas
narkotika
narkotikas
narkotika
narkotikans
narkotik_ans
kuriosum
kurios_um
kuriosumet
kurios_um_et
kuriosa
kuriosa
kuriosums
kurios_ums
kuriosumets
kurios_um_ets
kuriosas
kuriosa
sedativum
sedativ_um
sedativumet
sedativ_um_et
sedativet
sedativ_et
sedativa
sedativa
sedativums
sedativ_ums
sedativumets
sedativ_um_ets
sedativets

sedativ_ets
sedativas
sedativa
examen
exam_en
examina
exam_ina
examens
exam_ens
examinas
exam_inas
agens
agens
agenset
agens_et
agentier
agen_ti_er
agentierna
agen_ti_er_na
agensets
agens_ets
agentiers
agen_ti_ers
agentiernas
agen_ti_er_nas
prestanda
prestanda
prestandas
prestanda
star
star
staren
star_en
stars
stars
stars
stars
starens
star_ens
broiler
broiler
broilern
broilern
broilers
broilers
broilers
broilers
broilerns
broilern
broilers
broilers
broilers
broilers
fullmäktig
fullmäktig
fullmäktigen
fullmäktig_en
fullmäktige
fullmäktige

fullmäktigs
fullmäktig
fullmäktigens
fullmäktig_ens
fullmäktiges
fullmäktige
kammare
kam_mare
kammaren
kam_mar_en
kammarn
kam_marn
kamrar
kam_rar
kammarna
kam_mar_na
kamrarna
kam_rar_na
kammares
kam_mares
kammarens
kam_mar_ens
kammarns
kam_marns
kamrars
kam_rars
kammarnas
kam_mar_nas
kamrarnas
kam_rar_nas
pyjamas
pyjamas
pyjamas
pyjamas
pyjamasen
pyjamas_en
pyjamasar
pyjamas_ar
pyjamasarna
pyjamas_ar_na
pyjamasens
pyjamas_ens
pyjamasars
pyjamas_ars
pyjamasarnas
pyjamas_ar_na
s
sprej
sprej
sprejen
sprej_en
sprejer
sprej_er
sprejar
sprej_ar
sprejerna
sprej_er_na
sprejarna
sprej_ar_na
sprejs
sprej
sprejens
sprej_ens
sprejers
sprej_ers
sprejars
sprej_ars
sprejernas
sprej_er_nas
sprejarnas
sprej_ar_nas
gryn
gryn
grynet
gryn_et
grynen
gryn_en
grynerna
gryn_er_na
gryns
gryn
grynets
gryn_ets
gryners
gryn_ers
grynens
gryn_ens
grynernas
gryn_er_nas
koagel
koag_el
koaglen
koag_len
koagels
koag_els
koagels
koag_el
koaglens
koag_lens
medium
medi_um
mediumet
medi_um_et
mediet
medi_et
medier
medi_er
media

media
medierna
medi_er_na
mediums
medi_ums
mediumets
medi_um_ets
mediets
medi_ets
mediers
medi_ers
medias
media
mediernas
medi_er_nas
spektrum
spektr_um
spektret
spektr_et
spektrer
spektr_er
spektra
spektra
spektrerna
spektr_er_na
spektrana

spektra_na
spektrums
spektr_ums
spektrets
spektr_ets
spektrers
spektr_ers
spektras
spektra
spektrernas
spektr_er_nas
spektranas
spektra_nas
minimum
mini_mum
minimumet
mini_mum_et
minimet
mini_met
minima
mini_ma
minimums
mini_mums
minimumets
mini_mum_ets
minimets
mini_mets
minimas
mini_ma
serum
ser_um
serumet
ser_um_et
sera
sera
serums
ser_ums
serumets
ser_um_ets
seras
sera
votum
vot_um
votumet
vot_um_et
vota
vota
votums
vot_ums
votumets
vot_um_ets
votas
vota
forum
for_um
forum
for_um
forumet
for_um_et
fora
fora
forums
for_ums
forumets
for_um_ets
foras
fora
centrum
centr_um
centrumet
centr_um_et
centret
cent_ret
centra
centra
centrer
centr_er
centrums
centr_ums
centrumets
centr_um_ets
centrets
cent_rets
centras
centra
centrers
centr_er
tema
tema
temat
temat
teman
teman
temata
tema_ta
temas
tema
temats
temat
temans
teman
tematas
tema_tas
distikon
distik_on
distikon
distik_on
distikonet
distik_on_et
distika
distika
distikons
distik_on
distikons
distik_ons
distikonets
distik_on_ets
distikas
distika
franc
franc
francen
franc_en
francen
franc_en
francs
franc
francs
franc
francsen
franc_en
francens
franc_ens
francens
franc_ens
francsens
franc_ens
partner
partner
partnern
partnern
partners
partner
partners
partners
partnerns
partnern
partnerns
partnern
western
western
westerns
westerns
westerns
westerns
westerns
westerns

westerns	altares	huvud_enas	akne	brandväsens
hobby	altare	brus	akne	brand_väsens
hobby	altarets	brus	aknen	brandväsendets
hobbyn	altar_ets	bruset	aknen	brand_väsen_d
hobbyn	altarens	brus_et	aknes	ets
hobbies	altar_ens	brusets	akne	januari
hobbies	altarnas	brus_ets	aknens	januari
hobbyer	altar_nas	lymfa	aknen	januaris
hobby_er	tempo	lymfa	jubel	januari
hobbyerna	tempo	lymfan	jubel	bevåg
hobby_er_na	tempot	lymf_an	jublet	bevåg
hobbys	temp_ot	lymfas	jub_let	bevågs
hobby	tempon	lymfa	jubels	bevåg
hobbyns	temp_on	lymfans	jubel	ragnarök
hobbyn	tempi	lymf_ans	jublets	ragna_rök
hobbyers	tempi	exil	jub_lets	ragnaröks
hobby_ers	tempos	exil	andrum	ragna_röks
hobbyernas	tempo	exilen	andrum	alerten
hobby_er_nas	tempots	exil_en	andrummet	alerten
kilo	temp_ots	skam	andrum_met	fjolåret
kilo	tempons	skam	andrums	fjol_året
kilo	temp_on	skammen	andrum	fjolårets
kilo	tempis	skamm_en	andrummets	fjol_årets
kilot	tempi	skams	andrum_mets	fig
kilot	stimulus	skams	barium	fig
kilon	stimul_us	skammens	bari_um	nr
kilon	stimuluset	skamm_ens	bariumet	nr
kilona	stimul_us_et	början	bari_um_et	Sverige
kilo_na	stimuli	början	bariet	NIL
kilos	stimuli	början	bari_et	Sveriges
kilo	stimulusets	början	bariums	NIL
kilos	stimul_us_ets	praxis	bari_ums	Per
kilo	stimulis	praxis	bariumets	NIL
kilots	stimuli	praxisen	bari_um_ets	Pers
kilot	huvud	praxis_en	bariets	NIL
kilonas	huvud	praxisens	bari_ets	Alperna
kilo_nas	huvudet	praxis_ens	vimmel	NIL
kilos	huvud_et	passivum	vim_mel	Alpernas
kilo	huvuden	passiv_um	vimlet	NIL
kilos	huvud_en	passivums	vim_let	ABF
kilo	huvudena	passiv_ums	vimmels	NIL
altare	huvud_ena	kaffe	vim_mels	ABFs
altare	huvuds	kaffe	vimlets	NIL
altaret	huvud	kaffet	vim_lets	af
altar_et	huvudets	kaffet	brandväsen	NIL
altaren	huvud_ets	kaffes	brand_väsen	
altar_en	huvudens	kaffe	brandväsendet	
altarna	huvud_ens	kaffets	brand_väsen_d	
altar_na	huvudenas	kaffet	et	

Appendix B: Test Verbs and their Hyphenations

This is the output of the final test-run, November 19:th, of the file verb.test through the avstav function. The test word comes first, then the morphologically hyphenated version or if this is not possible a plus sign (+) which means it should be mechanically hyphenated.

bedja	bed_ja	bad	+	bedjande

bed_jan_de
bedjandes
bed_jan_des
bed
+
be
+
be
+
ber
NIL
bett
+
bett
+
beddes
be_dd
bads
+
beds
+
bedjas
bed_jas
begraven
begrav_en
begravna
begrav_na
bita
+
biter
bit_er
bet
+
bitit
bit_it
biten
bit_en
bitna
bit_na
bitande
bit_an_de
bit
+
bitandes
bit_an_des
bits
+
bets
+
bitits
bit_its
bli
+
bli
+
bliva
+
blir
+
bliver
bliv_er
blev
+
blivit
bliv_it
bliven
bliv_en
blivna
bliv_na
blivande
bliv_an_de
bragte
+
brinna
+
brinner
brinn_er
brann
+
brunnit
brunn_it
brunnen
brunn_en
brunna
brunn_na
brinnande
brinn_an_de
brista
+
brister
brist_er
brast
+
brustit
brust_it
brusten
brust_en
brustna
brust_na
bristande
brist_an_de
brist
+
brottas
brott_as
brottas
brott_as
brottas
brott_as
brottades
brott_ades
brottats
brott_ats
brottandes
brott_andes
bära
+
bär
+
bär
+
bar
+
burit
bur_it
buren
bur_en
burna
bur_na
bärande
bär_an_de
bärandes
bär_an_des
bäras
bär_as
bärs
+
bars
+
burits
bur_its
erfara
+
erfar
+
erfar
+
erfor
+
erfarit
erfar_it
erfaren
erfar_en
erfarna
erfar_na
erfarande
erfar_an_de
erfaras
erfar_as
erfars
+
erfarits
erfar_its
fanns
+
finnas
finn_as
finnas
finn_as
finns
+
finns
+
funnits
+
födas
+
födas
föd_as
föds
+
föds
+
föddes
+
föddes
+
föddes
föd_des
fötts
fö_ts
fötts
fö_ts
fött
+
fött
+
föd
+
födande
+
födda
+
föra
+
för
+
för
+
förde
för_de
förde
för_de
fört
+
fört
+
förande
för_an_de
förd
+
föras
för_as
förs
+
fördes
för_des
fördes
för_des
förts
för_ts
förandes
för_an_des
förfaras
förfar_as
förfars
+
ge
+
ge
+
ger
+
gav
+
gett
+
gett
+
given
giv_en
givna
giv_na
givande
giv_an_de
ges
+
gavs
+
getts
+
gick
+
giva
+
giver
giv_er
gav
+
givit
giv_it
given
giv_en
givna
giv_na
givande
giv_an_de
giv
+
givas
giv_as
givs
+
gavs
+
givits
giv_its
- gladde
gladdes
gladd_es
gladdes
gladd_es
glatt
+
glatt

+	ha_de	lyft	+	skämmas
glädja	haft	+	ser	skäm_mas
gläd_ja	+	lyft	NIL	skämmas
glädjer	has	+	såg	skäm_mas
gläd_jer	+	lyftande	+	skäms
- gladde	hava	lyft_an_de	sett	+
glatt	+	lyftas	+	skäms
+	havande	lyft_as	sett	+
glatt	hava_nde	lyftes	+	skämdes
+	heta	lyft_tes	seende	skäm_des
glädjande	+	lyftes	se_en_de	skämdes
gläd_jan_de	heter	lyft_es	sedd	skäm_des
gläd	het_er	lyfts	+	skämdes
+	hette	lyft_ts	ses	skäm_des
glädjas	het_te	lägga	+	skämts
gläd_jas	hetat	+	ses	skäm_ts
glädjas	+	lägger	+	skämts
gläd_jas	klä	lägg_er	sågs	skäm_ts
gläds	+	lade	+	skämd
+	klä	la_de	setts	+
glatts	+	lagt	+	skäms
+	kunna	+	skilja	+
glatts	+	lagt	+	skäms
+	kan	+	skiljer	+
gå	+	lagd	skilj_er	slå
+	kunde	+	skilde	+
gå	kun_de	lägg	skil_de	slå
+	kunnat	+	skilde	+
går	+	läggas	skil_de	slår
+	kunnande	lägg_as	skilt	+
gått	kun_an_de	läggs	+	slog
+	kännas	+	skilt	+
- ;gången	känn_as	lades	+	slagit
- ;gångna	kännas	la_des	skild	slag_it
göra	kän_as	lagts	+	slagen
+	känns	la_ts	skiljande	slag_en
gör	+	må	skilj_an_de	slående
+	kändes	+	skilj	slå_en_de
gör	kän_des	må	+	slås
+	kändes	+	skiljas	+
gjorde	kän_des	må	skilj_as	slås
gjor_de	kändes	+	skiljas	+
gjorde	kän_des	måtte	skilj_as	slåss
gjor_de	känts	+	skiljs	+
gjort	kän_ts	måste	+	slåss
+	legat	+	skiljs	+
gjort	+	måste	+	slåss
+	legat	+	skiljdes	+
gjord	+	måst	skilj_des	slogs
+	lyfta	+	skiljdes	+
görande	lyft_tta	nypt	skilj_des	slogs
gör_an_de	lyfta	+	skiljts	+
ha	+	nypt	skilj_ts	slagits
+	lyfter	+	skilts	slag_its
ha	lyft_er	obs	skil_ts	slagits
+	lyfte	+	skilts	+
har	lyft_te	se	skil_ts	spara
+	lyft	+	skolat	+
hade	+	se	+	spara

+	stå	stängas	+	sättande
sparar	+	stäng_as	svors	sätt_an_de
spar_ar	stå	stängs	+	sätt
sparade	+	+	svurits	+
spara_de	står	stängdes	svur_its	sättas
sparade	+	stäng_des	sy	sätt_as
spara_de	stod	stängdes	+	sätts
sparat	+	stäng_des	sy	+
spar_at	stått	stängts	+	sattes
sparat	+	stäng_ts	syr	+
spar_at	stått	stödja	+	satts
sparande	+	stöd_ja	sydde	+
spar_ande	- ;ståndеn	stödjer	sy_dde	söka
sparad	stående	stöd_jer	sydde	+
+	stå_en_de	stödjande	sy_dddde	söker
spar	ståendes	stöd_jan_de	sytt	sök_er
+	stå_en_des	stödjas	+	sökte
spar	stämma	stöd_jas	sytt	sök_te
+	stäm_ma	stöda	+	sökt
sparas	stämmer	+	sydd	+
spar_as	stäm_mer	stöder	+	sökt
sparas	stämde	stöd_er	sydda	+
spar_as	stäm_de	stödde	sy_dddda	sökta
sparades	stämde	stöd_de	sys	sök_tta
spara_d	stäm_de	stödde	+	sökande
sparades	stämt	stöd_de	syddes	sök_an_de
spara_des	+	stött	sy_ddes	sökandes
sparats	stämt	+	syddes	sök_an_des
spar_ats	+	stött	sy_dd	sökas
spars	stämd	+	sytts	sök_as
+	+	stödd	+	söks
spänna	stämmas	+	synas	+
+	stäm_mas	stöd	syn_as	söktes
spänner	stäms	+	syns	sök_tes
spänn_er	+	stöds	+	sökts
spände	stämdes	+	syntes	sök_ts
spän_de	stäm_des	stöddes	syn_tes	sök
spände	stämdes	stöd_des	synts	+
spän_de	stäm_des	stöddes	syn_ts	ta
spänt	stämts	stöd_des	säja	+
+	stäm_ts	stötts	+	ta
spänt	stänga	+	säjer	+
+	+	svära	säj_er	tar
spänd	stänger	+	sa	+
+	stäng_er	svär	+	tagit
spännande	stängde	+	sade	tag_it
spänn_an_de	stäng_de	svär	sa_de	tagande
spänn	stängde	+	sagt	tag_an_de
+	stäng_de	svor	+	tagen
spännas	stängt	+	sagt	tag_en
spänn_as	+	svurit	+	tagna
spänns	stängt	svur_it	sätta	tag_na
+	+	svuren	+	tas
spändes	stängd	svur_en	sätter	+
spän_des	+	svurna	sätt_er	tas
spändes	stängande	svur_na	satte	+
spän_des	stäng_an_de	svärande	+	tas
spänts	stäng	svär_an_de	satt	+
spän_ts	+	svärs	+	togs

+	tydas	tän_ts	vetandes	+
togs	+	tordas	vet_an_des	älskar
+	tända	tor_as	vore	älsk_ar
tagits	tän_da	törs	+	älskade
tag_its	tända	+	välja	älsk_ade
tagits	+	tordes	+	älskade
tag_its	tänder	tor_des	väljer	älsk_ade
tags	tänd_er	torts	välj_er	älskat
+	tände	tor_ts	valde	älsk_at
tog	tän_de	vara	val_de	älskat
+	tände	+	valde	älsk_at
trädde	tän_de	är	val_de	älskad
+	tänt	+	valt	älsk_ad
trädde	+	var	+	älskade
+	tänt	+	valt	älsk_ade
trätt	+	var	+	älskade
+	tändande	+	vald	älsk_ade
trätt	tänd_an_de	varit	+	älskande
+	tänd	var_it	väljande	älsk_ande
tyda	+	varande	välj_an_de	älskandes
+	tänd	var_an_de	väljas	älsk_andes
tyder	+	varandes	välj_as	älskas
+	tändas	var_an_des	väljs	älsk_as
tydde	tän_das	veta	+	älskas
+	tändas	+	valdes	älsk_as
tydde	tänd_as	vet	val_des	älskades
+	tänds	+	valdes	älsk_ad
tytt	+	vet	val_des	älskades
+	tändes	+	valts	älsk_ades
tytt	tän_des	visste	val_ts	älskats
+	tändes	+	vämjas	älsk_ats
tydd	tän_des	vetat	vämj_as	
+	tändes	+	älska	
tyd	tän_des	vetande	+	
+	tänts	vet_an_de	älska	

Appendix C: Test Adjectives and their Hyphenations

This is the output of the final test-run, December 6:th, of the file adj.test through the avstav function. The test word comes first, then the morphologically hyphenated version or if this is not possible a plus sign (+) which means it should be mechanically hyphenated.

blek	blekaste	friaste	hög	högsta
+	blek_aste	fri_aste	+	hög_sta
blekt	fri	lätt	högt	ung
+	+	+	+	+
bleke	fritt	lätta	höga	ungt
+	+	+	+	+
bleka	fria	lättare	högre	unge
+	+	lätt_are	hög_re	+
blekare	friare	lättast	högst	unga
blek_are	fri_are	lätt_ast	hög_st	+
blekast	friast	lättaste	högste	yngre
blek_ast	fri_ast	lätt_aste	hög_ste	yng_re

yngst
yng_st
yngsta
yng_sta
yngste
yng_ste
hård
+
hårt
+
hårda
+
hårdare
hår_are
hårdast
hår_ast
hårdaste
hår_aste
glad
+
glatt
+
glatt
+
glatt
+
glada
+
gladare
glad_are
gladast
glad_ast
gladaste
glad_aste
konstlad
NIL
konstlad
+
konstlat
+
konstlade
konstla_de
vattenkyld
+
vattenkylt
+
vattenkylda
+
högljudd
+
högljutt
+
högljudda
+
högljuddare
högljudd_are
högljuddast
högljudd_ast
högljuddaste
högljudd_aste
datorstödd
+
datorstött
+
datorstödda
+
tunn
+
tunt
+
tunna
+
tunnare
tunn_are
tunnast
tunn_ast
tunnaste
tunn_aste
ensam
+
ensamt
+
ensamma
ensam_ma
ensammare
ensam_mare
ensammast
ensam_mast
ensammaste
ensam_maste
ädel
+
ädelt
+
ädla
äd_la
ädlare
äd_lare
ädlast
äd_last
ädlaste
äd_laste
vacker
+
vackert
+
vackra
vack_ra
vackre
vack_re
vackrare
vack_rare
vackrast
vack_rast
vackraste
vack_raste
vänster
+
vänstra
+
angelägen
angeläg_en
angeläget
angeläg_et
angelägna
angeläg_na
angelägnare
angeläg_nare
angelägnast
angeläg_nast
angelägnaste
angeläg_naste
vattenburen
vattenbur_en
vattenburet
vattenbur_et
vattenburna
vattenbur_na
välfunnen
välfunn_en
välfunnet
välfunn_et
välfunna
+
ljummen
ljum_men
ljummet
ljum_met
ljumna
ljum_na
ringa
+
ringare
ring_are
ringast
ring_ast
ringaste
ring_aste
enda
+
ende
+
bra
+
bra
+
bättre
bätt_re
bättre
bätt_re
bäst
+
bäst
bä_st
bästa
bä_sta
bäste
bä_ste
gammal
+
gammalt
+
gamla
+
äldre
äld_re
äldst
äld_st
äldsta
äld_sta
äldste
äld_ste
liten
+
litet
+
lilla
+
lille
+
små
+
minst
min_st
minst
min_st
minsta
min_sta
minste
min_ste
närmare
närm_are
närmast
närm_ast
närmaste
närm_aste
bakre
bak_re
bakerst
bak_erst
bakersta
bak_ersta
fler
+
flera
+
flest
+
flesta
fles_ta
mer
+
mer
+
mera
me_ra
mest
+
mest
+
mesta
me_sta
meste
me_ste
gängse
+
vilse
+
stackars
+
norra
+
samtliga
+
små
+
mindre
mind_re
mindre
mind_re
färre
+
aktra
+
mellersta
meller_sta

Appendix D: Test Adverbs and their Hyphenations

This is the output of the final test-run, December 6:th, of the file adverb.test through the avstav function. The test word comes first, then the morphologically hyphenated version or if this is not possible a plus sign (+) which means it should be mechanically hyphenated.

ofta	redan	bästa	mest	vidare
+	+	bä_sta	+	vid_are
oftare	bra	bättre	mest	mitterst
oft_are	+	bätt_re	+	mitt_erst
oftast	bra	bättre	mindre	mittersta
oft_ast	+	bätt_re	mind_re	mitt_ersta
fort	bäst	mer	mindre	längre
+	+	+	mind_re	läng_re
fortare	bäst	mer	minst	längst
fort_are	bä_st	+	min_st	läng_st
fortast	bäste	mera	minst	
fort_ast	bä_ste	me_ra	min_st	