Alfred Holl

Compact reverse dictionaries on the inflectional morphology of verbs as interface between lexicon and grammar

1 Motivation, starting point
2 Research issues
3 Research method: data mining
4 Research results
5 The English and the German verb
6 Benefit of the results
7 Future work
8 Examples to be presented
9 Literature

1 Motivation, starting point: observation

Mistakes in the language of children
Hypercorrect forms of native speakers
Mistakes of non-native speakers / language learners
e.g.

*he *readed*	analogous to *he pleaded*
*think, *thank, *thunk*	analogous to *sink, sank, sunk*
*er *tretet*	analogous to *er knetet, betet*
**getretet*	analogous to *geknetet, gebetet*
*er hat *gelügt*	analogous to *er hat gerügt*
**gestreitet*	analogous to *ausgebreitet*
*ich habe *gewerft*	analogous to *ich habe gekauft*
*du *sehst*	analogous to *du stehst*
*er *frägt*	analogous to *er trägt*

1 Motivation, starting point: explanation

Which strategy (of learning) leads to these mistakes?
The non-reflected use of the assumption of morphologic analogy
(*morphology* here always used in the sense of *inflectional morphology*)

The reverse similarity of the infinitive (lexical base)
shall be used as key feature (gestalt-psychological term)
for the assumption of morphologic analogy,
that is, for the assumption of the equality of
all the essential morphologic features, that is,
stem alternations and endings in synthetic forms.

Idea:
use the known averbo of a pattern lexeme as paradigm for another lexeme
in order to “derive” its (unknown) averbo.

1 Motivation, starting point: explanation

But the assumption of analogy (Analogieschluss, Schluss auf Analogie) is only a heuristic method and does not necessarily lead to correct results.

Reversely similar lexemes (short for *lexemes with reversely similar lexical bases*) can have equal or different morphological properties.

True assumptions of morphologic analogy:

~ling (~lung, ~lung)*: cling, fling, sling*

~reiben (~rieb ge~rieben)*: reiben, schreiben, treiben*

1 Motivation, starting point: explanation

False assumptions of morphologic analogy:

~ch*: hatch, match, watch, fetch* are reg.
teach (taught taught), catch (caught, caught)

~eiben: to *-reiben* add *einverleiben (-te, -t), bleiben*

~iegen*:*
biegen (bog gebogen), fliegen, wiegen
liegen (lag gelegen)
siegen (siegte gesiegt), kriegen, schmiegen

1 Motivation: further reasons for dealing with reverse similarity

1 A basic verb (*simplex*) and its prefixed verbs
have reversely similar infinitives and
in the majority of cases the same conjugation.

2 Only regular inflection types are productive
(partly with phonotactic, phonetic or orthographic particularities):
new lexemes (neologisms) will never be irregular, except for fun.
The inflection type is assigned to new lexemes
via reverse similarity to existing lexemes.

3 In Latin and the Romance languages,
the assignment to conjugation classes
is based on the infinitive ending.

1 Motivation: further reasons for dealing with reverse similarity

4 Phonotactically, phonetically and orthographically caused specialties of regular verbs can be seen at the infinitive ending.
The corresponding inflection subtypes are productive.

consonant grapheme doubling (*digging*)
e-insertion in the 3rd singular of present tense for English verbs ending in *–ch, –sh, –Co, –ss, –x, –Cz* (*wishes*)
e-deletion (*aging*)
ie-*y* change (*dying*)
y-*ie* change (*studies*)

e-Einschübe bei Verben auf -*den*, -*ten*, -*Cmen* (*C* ungleich *h*, *l*, *m*, *r*; alle homogen), -*Cnen* (nur -*fnen*, -*gnen*, -*chnen* homogen)
s-Elision im Ind.Prs.2.Sg bei Verben auf -*sen*, -*ssen*, -*ßen*, -*xen* (hom.), -*zen*

1 Motivation: further reasons for dealing with reverse similarity

5 General tendency to simplify inflectional systems
Irregular forms are only then very stable if they occur very often.
Otherwise: (winning) regular and irregular forms in parallel,
partly with semantic or diaphasic differentiation

learn: learned / learnt; lie: lied / lay, lied / lain

stieben: stiebte / stob
weben: webte / wob
senden: sendete (Radio, TV) / *sandte* (Brief)
wenden: wendete / wandte
triefen: triefte / troff
schaffen: schaffte (‘fertigbringen’) / *schuf* (‘erschaffen’)
pflegen: pflegte (gängig) / *pflog* (obsolet)
glimmen: glimmte / glomm

2 Research issues

Identify and classify different types of groups (clusters)
with (only or mostly morphologically analogous) reversely similar lexemes

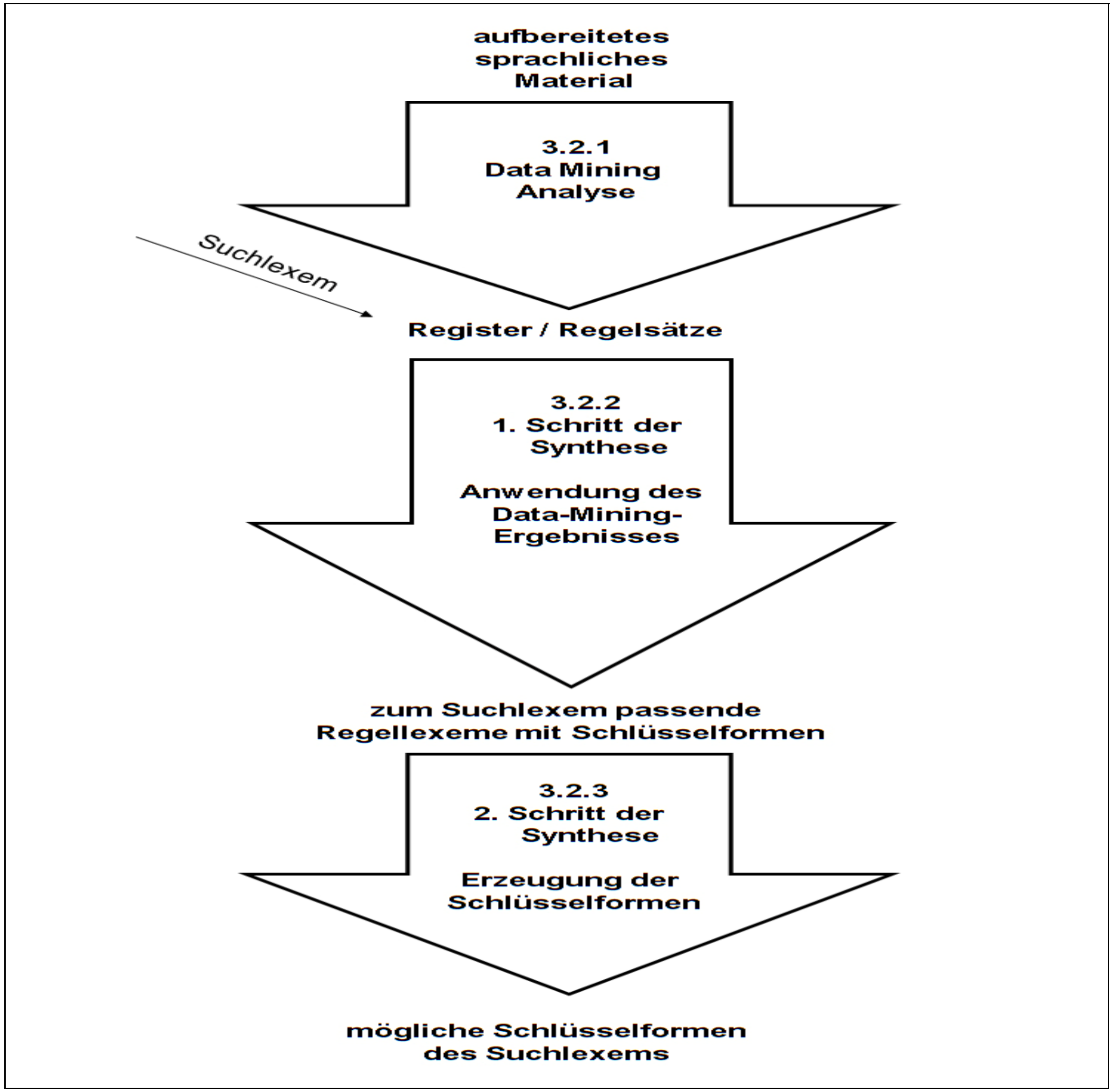
Examine, systematize and structure the relation
between the reverse similarity of lexemes and their morphologic analogy
(when do both coincide?)
in various language-part-of-speech combinations

Design an analytic algorithm to do this task (data mining)

Publish the results in the form of compact dictionaries

Design a search algorithm
which finds the inflection type of an arbitrary lexeme
using the adequate dictionary.

3 Research method: data mining (process)



3 Research method: data mining (process)

3.1 Pre-processing

Representation of linguistic objects (lexemes)
- graphemic

Modified orthographic conventions
- prefix treatment (e.g. German *erben*), marking of inflectional variants

Sources for gathering linguistic data
- reverse dictionaries, verb lists, descriptive grammars

Delimitations in comparison to syntax and lexicon
- no analytic forms and no details of defective verbs

Marking inflectional features:
- principal parts (1st: lexical base) and inflection types (identifier)

Derivation rules and exceptions
- construction of rules, list of not derivable forms

3.2 Processing: top-down cluster analysis

Design and execution of the data mining algorithm
independent of the examined language-part-of-speech combination
in a inflecting or agglutinating language

3.3 Post-processing

Interpretation, evaluation of the data mining result
Preparation for use, typographic marking
Consistent definition of
- homogeneous clusters
- basic clusters
- cluster tree

Design of a search algorithm
independent of the examined language-part-of-speech combination
which also produces the principal parts of the arbitrary verb

4. Research results

Homogeneous cluster (homC)
Mathematically connected set (no gaps or interruptions)
in a reversely ordered sequence of morphologically analogous lexemes

Basic cluster (basC) – inhomogeneous
– most of the verbs have the same morphological properties
– threshold percentage of the majority: around 70%
– often correspond to traditional inflection classes

Trivial homogeneous clusters
One-lexeme homogeneous clusters with implicit word delimitation
They consist of one single basic verb and its prefixed verbs.
They have to be dealt with from a formal perspective
although they are not interesting from a linguistic one.
Examples: *dig (dug, dug)*; *bring (brought, brought)* etc.

4.1 / 2 Homogeneous and basic clusters

1 Many-lexeme clusters,
without grapheme type characters and without word delimitation

Homogeneous clusters	Basic clusters
Example: *~ling (~lung, ~lung)* All verbs whose infinitives end in *–ling*. All of them have the same irregularities. Elements: *cling (clung, clung); fling (flung, flung); sling (slung, slung)*	Example: *~ch (~ched, ~ched)* All verbs with infinitive ending *–ch*. Mostly regular with *e*-insertion in the 3rd sg present tense whose ending is extended to *–es*. Elements: *reach (reached, reached)* etc. Exceptions: *teach (taught, taught); catch (caught, caught)*

4.1 / 2 Homogeneous and basic clusters

1 Many-lexeme clusters,
without grapheme type characters and without word delimitation

Homogeneous clusters	Basic clusters
Example: *~reiben (~rieb, ge~rieben)* All verbs whose infinitives end in *–reiben*. All of them have the same irregularities. Elements: *reiben (rieb gerieben), schreiben, treiben*	Example: *~den (~dete, ge~et)* All verbs with infinitive ending *–den*. Mostly regular with *e*-insertion in the 2nd sg, 3rd sg, 2nd pl present tense whose endings are extended to *–est, –et, –et*. Elements: *baden (badete, gebadet)* etc. Exceptions: *winden (wand, gewunden)* etc.

4.1 / 2 Homogeneous and basic clusters

2 Many-lexeme clusters (precisely: sets of clusters) with grapheme type characters (*C*, *V*) and without word delimitation

Homogeneous clusters	Basic clusters
Example: *~Cz (~Czed, ~Czed)* All verbs whose infinitives end in a consonant grapheme plus z. All of the verbs are regular with *e*-insertion in the 3rd sg present tense whose ending is extended to *–es*. Elements: *jazz (jazzed, jazzed)* etc.	Example: *~C (~Ced, ~Ced)* All verbs whose infinitives end in a consonant grapheme. Mostly regular Elements: *look (looked, looked)* etc. Exceptions: *bring (brought, brought)* etc. There are many exceptions, but statistically only a few as the cluster comprises all of the regular verbs ending in a consonant grapheme.

4.1 / 2 Homogeneous and basic clusters

2 Many-lexeme clusters (precisely: sets of clusters) with grapheme type characters (*C*, *V*) and without word delimitation

Homogeneous clusters	Basic clusters
Constructed example because not exactly the same inflection type: *~Cn (~Cte, ge~Ct)* All verbs whose infinitives end in a consonant grapheme plus n. All of the verbs are regular with exceptions in the present subjunctive. Elements: *handeln (handelte, gehandelt), zaubern (zauberte, gezaubert)* etc.	No adequate example in the German verbal system. The following two clusters are not recommended: basC2 *~Cmen* (reg / +e(mn)) would require basC3 *~hmen* (reg), homC *~lmen* (reg), basC3 *~mmen* (reg), homC *~rmen* (reg) basC2 *~Cnen* (reg / +e(mn)) would require homC *~Vhnen* (reg), basC3 *~nnen* (reg), homC *~rnen* (reg)

4.1 / 2 Homogeneous and basic clusters

3 Many-lexeme clusters (precisely: sets of clusters) with grapheme type characters and with explicit word delimitation

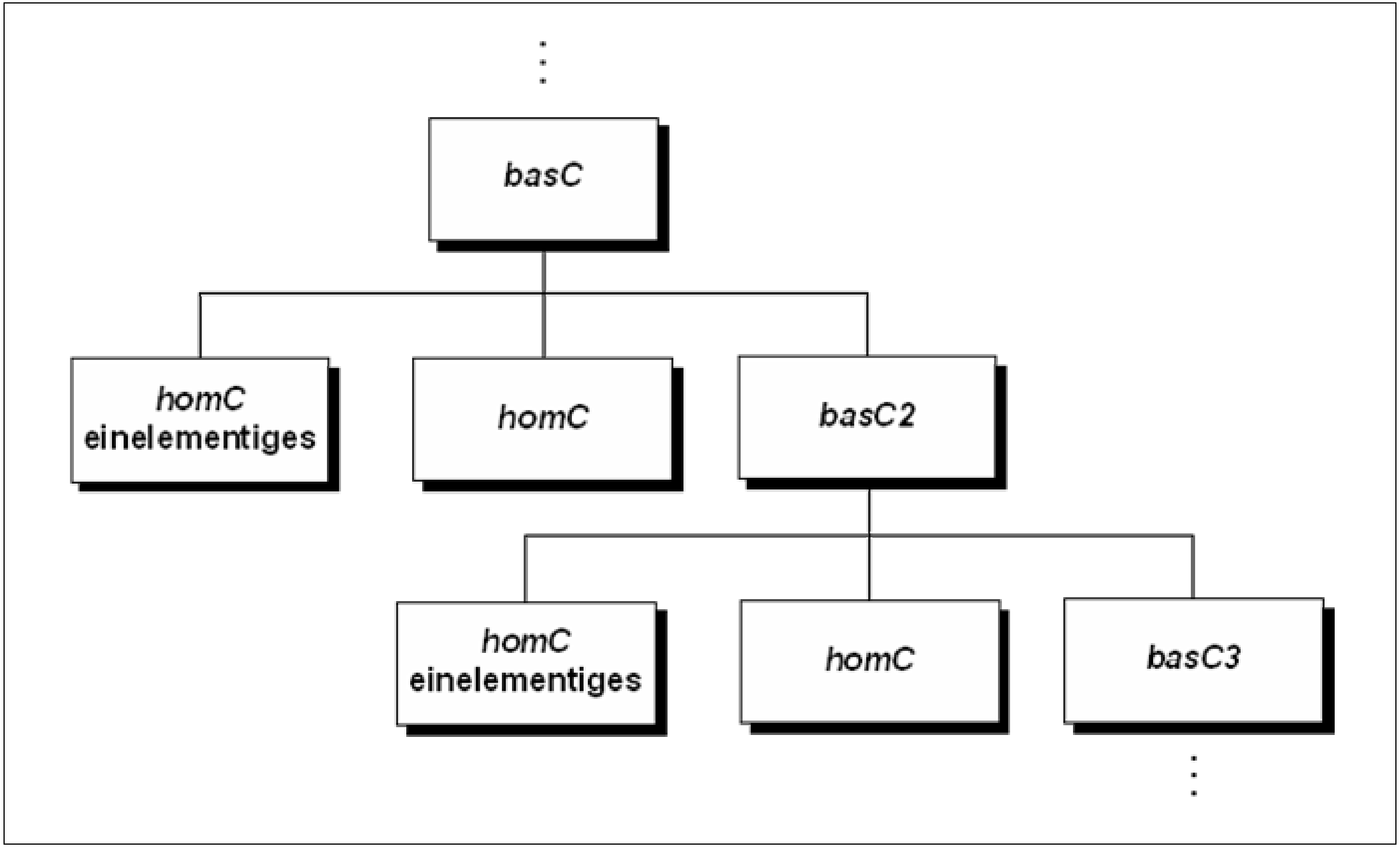
Homogeneous clusters	Basic clusters
Example: *#CCVg (#CCVgged, #CCVgged)* All basic verbs whose infinitives have the form "two consonant graphemes plus vowel grapheme plus *g*" and their prefixed verbs. All verbs are regular with consonant grapheme doubling in past tense, past participle and gerund. Elements: *drag (dragged, dragged), flag (flagged, flagged)* etc.	Example: *#CVg (#CVgged, #CVgged)* All basic verbs whose infinitives have the form "consonant grapheme plus vowel grapheme plus *g*" and their prefixed verbs. Mostly regular with consonant grapheme doubling in past tense, past participle and gerund. Elements: *jig (jigged, jigged)* etc. Exception: *dig (dug, dug)*

4.1 / 2 Homogeneous and basic clusters

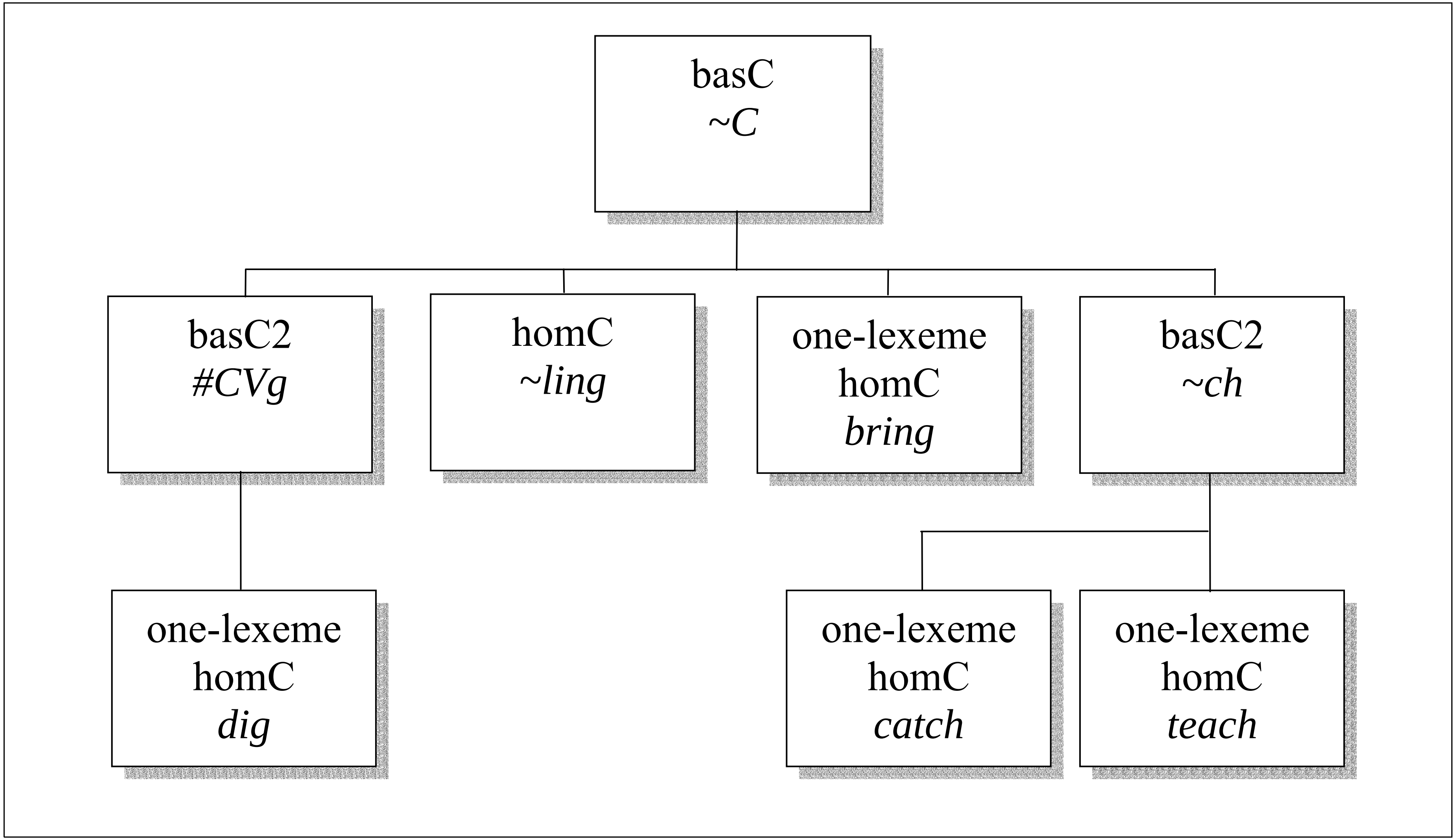
3 Many-lexeme clusters (precisely: sets of clusters) with grapheme type characters and with explicit word delimitation

Homogeneous clusters	Basic clusters
No example in the German verbal system	No example in the German verbal system

4.3 Cluster trees: general



4.3 Cluster trees: part of the English verbs



4.3 Well-defined cluster trees: conventions for cluster hierarchies

It must be possible
to unambiguously assign a search lexeme to its best-fit cluster.

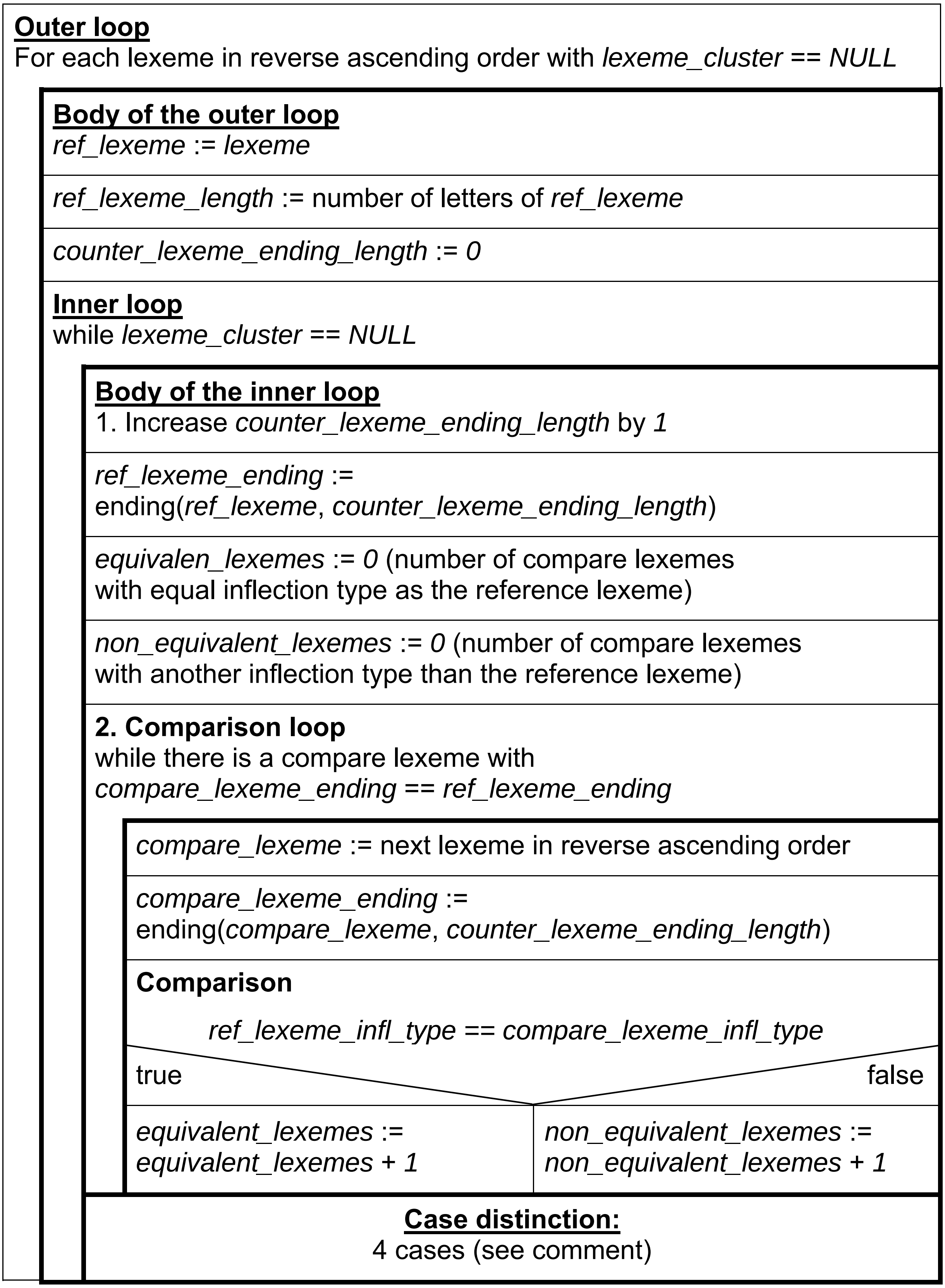
Two sets of lexemes belonging to two different clusters must be
1. either disjoint
2. or in a subset-superset relation.

The lexical base of a parent cluster always contains
fewer or as many characters than the ones of its child clusters.

In the case of an equal number of characters,
the parent cluster contains more grapheme type characters.

That is, regarding lexical bases (alphabetic properties) as conditions:
the condition of a parent cluster is always weaker
than the conditions of its child clusters.

4.4 Data mining algorithm



```
Outer loop
For each lexeme in reverse ascending order with lexeme_cluster == NULL
    Body of the outer loop
    ref_lexeme := lexeme
    ref_lexeme_length := number of letters of ref_lexeme
    counter_lexeme_ending_length := 0
    Inner loop
    while lexeme_cluster == NULL
        Body of the inner loop
        1. Increase counter_lexeme_ending_length by 1
        ref_lexeme_ending :=
        ending(ref_lexeme, counter_lexeme_ending_length)
        equivalen_lexemes := 0 (number of compare lexemes
        with equal inflection type as the reference lexeme)
        non_equivalent_lexemes := 0 (number of compare lexemes
        with another inflection type than the reference lexeme)
        2. Comparison loop
        while there is a compare lexeme with
        compare_lexeme_ending == ref_lexeme_ending
            compare_lexeme := next lexeme in reverse ascending order
            compare_lexeme_ending :=
            ending(compare_lexeme, counter_lexeme_ending_length)
            Comparison
                ref_lexeme_infl_type == compare_lexeme_infl_type
            true
                equivalent_lexemes :=
                equivalent_lexemes + 1
            false
                non_equivalent_lexemes :=
                non_equivalent_lexemes + 1
        Case distinction:
        4 cases (see comment)
```

4.4 Data mining algorithm

Top-down cluster analysis strategy (divisive method)

The database established in the pre-processing phase has to be sorted reversely using the column “analytic lexical base”.

The column “lexeme cluster” is initialized with NULLs. It is meant to be filled with the name of the cluster found by the algorithm.

The core of the algorithm (Fig.) consists of two nested loops:
step by step, the outer loop processes all of the lexemes;
step by step, the inner loop processes all ending lengths.

In detail, a general step of the algorithm runs as follows.

n-th step of the outer loop:

The next lexeme is picked out which is not assigned to a cluster, that is, which has the initial value in the column “lexeme cluster” (*lexeme_cluster == NULL*). This lexeme is called **reference lexeme** (variable *ref_lexeme*).

m-th step of the inner loop:

1. The variable *Counter_lexeme_ending_length* is increased by *1* and thus set to *m*. The algorithm gets the reference lexeme’s ending (*ref_lexeme_ending*, depending on the current value of *counter_lexeme_ending_length*) and its inflection type (**reference inflection type**). The algorithm is now going to compare the next lexemes with the reference lexeme. Therefore, the variables of the following comparison loop are initialized with *0*.

2. In a third loop (comparison loop), all of the lexemes which have the same *n*-digit ending (the same *n* trailing letters in the lexical base), are examined (**compare lexemes**). All of the compare lexemes with the reference inflection type are counted in the variable *equiv_lexemes*. Those with a different inflection type are counted in the variable *non_equiv_lexemes*.

3. Depending on the values of the variables *equiv_lexemes, non_equiv_lexemes* and *counter_lexeme_ending_length*, four cases are distinguished (Fig.).

4.4 Data mining algorithm

	equivalent_ lexemes	non_equivalent_ lexemes	counter_lexeme_ ending_length
Case 1	0	0	—
Case 2	—	>= 1	= ref_lexeme_length
Case 3	>= 1	0	—
Case 4	—	>= 1	< ref_lexeme_length

Cases after counting
equivalent and non-equivalent compare lexemes

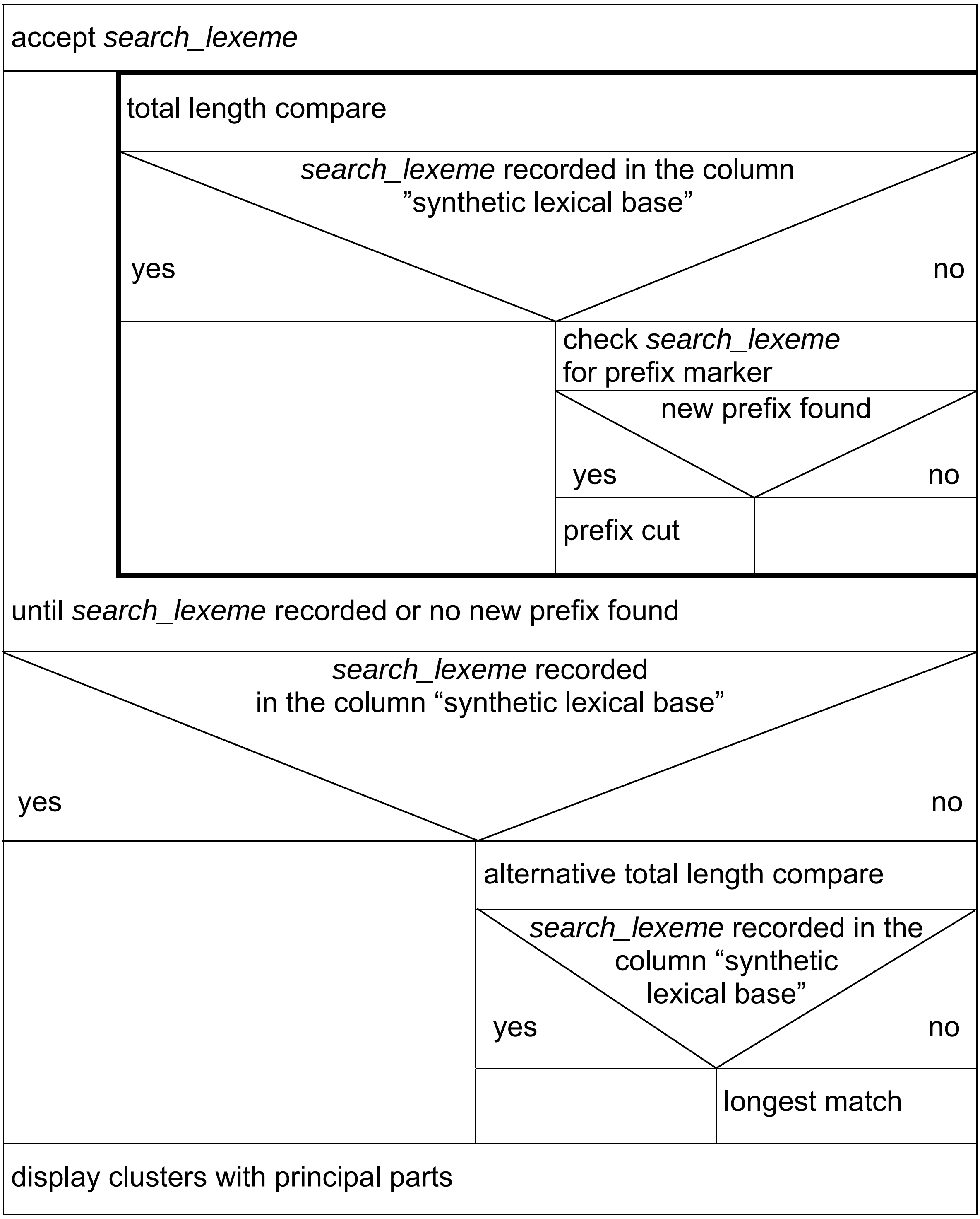
Case 1 (there are no compare lexemes)
one-lexeme cluster consisting of the reference lexeme only

Case 2 (there are 1 or more compare lexemes with another inflection type than the reference inflection type, and the current ending length is already equal to the number of letters of the reference lexeme)
one-lexeme cluster consisting of the reference lexeme only

Case 3 (all of the compare lexemes have the reference inflection type)
many-lexeme cluster consisting of the reference lexeme and at least than 1 more lexeme: many-lexeme homogeneous cluster

Case 4 (there are 1 or more compare lexemes with another inflection type than the reference inflection type,
and the current ending length is still less than the number of letters of the reference lexeme)
no cluster
The inner loop continues with an increase of the current ending length (*counter_lexeme_ending_length*) by *1*.

4.5 Search algorithm: the use of the data mining result



accept *search_lexeme*

total length compare

search_lexeme recorded in the column ”synthetic lexical base”

yes

no

check *search_lexeme* for prefix marker

new prefix found

yes

no

prefix cut

until *search_lexeme* recorded or no new prefix found

search_lexeme recorded in the column “synthetic lexical base”

yes

no

alternative total length compare

search_lexeme recorded in the column “synthetic lexical base”

yes

no

longest match

display clusters with principal parts

4.5 Search algorithm:
the use of the data mining result

Prefix cut cuts a prefix off the search lexeme.

Total length compare finds 1 cluster type:
one-lexeme homC.

Alternative total length compare creates
search alternatives 4 to 0,
the *n*-th by replacing normal by grapheme type characters
and conserving *n* trailing normal characters.

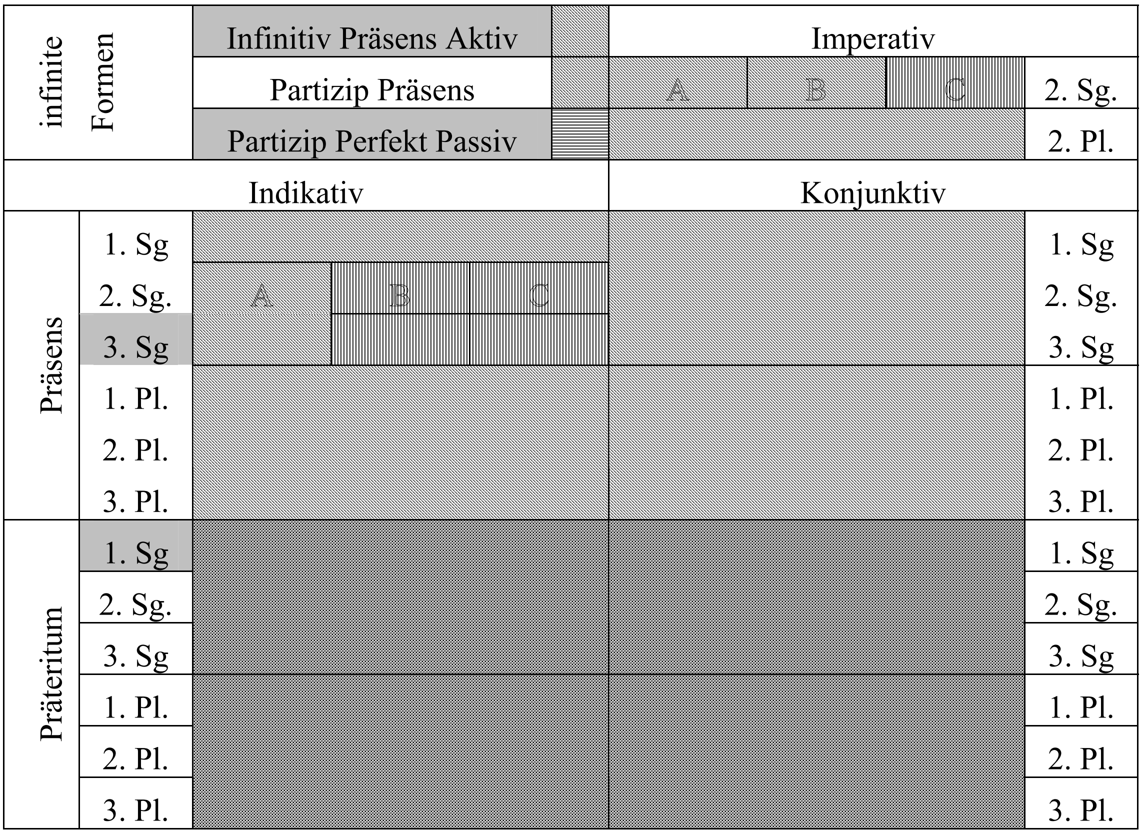
Alternative total length compare finds 2 cluster types:
homC and basC
with word delimitation and
grapheme type characters.

Longest match reduces
the search lexeme and its search alternatives
from the left by one character after the other.

Longest match finds 4 cluster types:
homC and basC
without word delimitation,
no matter whether with or without grapheme type characters.

5.D The situation of the German verb

5.D.1 Principal parts, stem distribution



infinite Formen		Infinitiv Präsens Aktiv			Imperativ			
		Partizip Präsens			A	B	C	2. Sg.
		Partizip Perfekt Passiv						2. Pl.
Indikativ					Konjunktiv			
Präsens	1. Sg							1. Sg
	2. Sg.	A	B	C				2. Sg.
	3. Sg							3. Sg
	1. Pl.							1. Pl.
	2. Pl.							2. Pl.
	3. Pl.							3. Pl.
Präteritum	1. Sg							1. Sg
	2. Sg.							2. Sg.
	3. Sg							3. Sg
	1. Pl.							1. Pl.
	2. Pl.							2. Pl.
	3. Pl.							3. Pl.

A: Indikativ Präsens 2. / 3. Sg. und Imperativ 2. Sg. leiten sich vom Infinitiv ab. Indikativ Präsens 3. Sg. ist keine Schlüsselform.

gehen – du gehst – er geht – geh!

B: Indikativ Präsens 2. Sg. leitet sich vom Indikativ Präsens 3. Sg. (Schlüsselform) ab, der Imperativ 2. Sg. aber vom Infinitiv.

lassen – du lässt – er lässt – lass!

C: Indikativ Präsens 2. Sg. und Imperativ 2. Sg. leiten sich vom Indikativ Präsens 3. Sg. ab, der als Schlüsselform genannt ist.

treffen – du triffst – er trifft – triff!

(Holl / Behrschmidt / Kühn 2004, 103-104)

5.D.2 Homogeneous clusters

Inflection type	Lex. base	Examples
ei-ie-ie	**~reiben**	**reiben, schreiben, treiben**
i-a-u / +e(dt)	**~winden**	**winden, schwinden**
i-a-u	**~lingen**	**schlingen, klingen, ge- / miss_lingen**
i-a-u	**~wingen**	**schwingen, zwingen**
ie-o-o	**~riechen**	**riechen, kriechen**
ei-i-i	**~leichen**	**bleichen, gleichen, schleichen**
reg / +e(mn)	~dmen	widmen
i-o-o	**~limmen**	**glimmen, klimmen**
reg / +e(mn)	~tmen	atmen
reg / +e(mn)	~bnen	ebnen
reg / +e(mn)	~dnen	ordnen
reg / +e(mn)	~fnen	öffnen
reg / +e(mn)	~gnen	eignen
reg / +e(mn)	~chnen	rechnen
reg / +e(mn)	~cknen	trocknen
reg / +e(mn)	~mnen	ver_vollkommnen
reg / +e(mn)	~pnen	wappnen
ie-o-o / -s / +e(sz) / +CC	**~ließen**	**fließen, schließen**
ie-o-o / -s / +e(sz) / +CC	**~rießen**	**sprießen, ver_drießen**
reg / -s(sz)	~xen	faxen
reg / -e(lr)	~eln	jubeln
reg / -e(lr)	~ern	zaubern

5.D.2 Homogeneous clusters

***~reiben**: reiben (rieb gerieben), schreiben, treiben*

***~winden**: winden (windet wand gewunden), schwinden*

***~lingen**: schlingen (schlang geschlungen), klingen, ge-/miss_lingen*

***~wingen**: schwingen (schwang geschwungen), zwingen*

***~riechen**: riechen (roch gerochen), kriechen*

***~leichen**: bleichen (blich geblichen)* (auch regelmäßig), *gleichen, schleichen*

***~limmen**: glimmen (glomm, geglommen)* (auch regelmäßig), *klimmen* (auch regelmäßig)

***~ließen**: fließen (floss geflossen), schließen*

***~rießen**: sprießen (spross gesprossen)* (auch regelmäßig), *verdrießen*

5.D.3 Inhomogeneous clusters 1

Necessarily contain irregular verbs

Selection of clusters with 3-4 inflection types

graben (grub, gegraben)
haben (hatte, gehabt)
laben (labte, gelabt), schaben, traben

schneiden (schnitt, geschnitten), leiden
meiden (mied, gemieden), scheiden
kleiden (kleidete, gekleidet)

liegen (lag gelegen)
biegen (bog gebogen), fliegen, wiegen
siegen (siegte gesiegt), kriegen, schmiegen

gehen (ging, gegangen)
stehen (stand, gestanden)
sehen (sah, gesehen), geschehen
flehen (flehte, gefleht), drehen

schwimmen (schwamm, geschwommen)
glimmen (glomm - glimmte, geglommen - geglimmt), klimmen
stimmen (stimmte, gestimmt), grimmen, trimmen

heißen (hieß, geheißen)
beißen (biss, gebissen), scheißen, schleißen, schmeißen, reißen
spleißen (spliss - spleißte, gesplissen, gespleißt)
weißen (weißte, geweißt), gleißen, kreißen, schweißen

etc.

5.D.3 Inhomogeneous clusters 2

Clusters with 2 inflection types only are considerably more frequent

~eiben (fast homogen): zu *~reiben* kommt hinzu *bleiben*; Ausnahme *einver-/ent_leiben* (regelmäßig)

~ingen (weitgehend homogen): zu *~lingen* und *~wingen* kommen hinzu *dingen* (auch regelmäßig), *ringen, dringen, springen, wringen, singen*; Ausnahmen *bringen (brachte gebracht), be-/um_ringen* (regelmäßig)

~ießen (fast homogen): zu *~ließen* und *~rießen* kommen hinzu *gießen, schießen, genießen*; Ausnahme *spießen* (regelmäßig)

etc.

5.D.4 Basic clusters

Cluster information	Basic cluster: morphological property (infl. type)	Basic cluster: alphabetic property (lexical base)	Examples	Exceptions
basC	reg	~en	loben	gehen
basC2 (er badet)	reg / +e(dt)	~den	baden	winden
basC2; no! caution	reg / +e(mn)	~Cmen	atmen	formen, kommen
basC2; no! caution	reg / +e(mn)	~Cnen	ebnen	lernen, kennen
basC2 (hat studiert)	reg / -ge(ieren)	~ieren	studieren	stieren
basC2 (du grast)	reg / -s(sz)	~sen	grasen, passen	blasen, essen
basC2 (du spaßt)	reg / -s(sz)	~ßen	spaßen	fließen
basC2 (er watet)	reg / +e(dt)	~ten	waten	raten
basC2 (du pflanzt)	reg / -s(sz)	~zen	pflanzen	sitzen

basC
~en (~t, ~te, ge~t)

basC2
~den (~det, ~dete, ge~det): baden
~Cmen (~Cmet, ~Cmete, ge~Cmet): atmen
~Cnen (~Cnet, ~Cnete, ge~Cnet): öffnen
~ieren (~iert, ~ierte, ~iert): studieren
~sen
~ßen
~ten (~tet, ~tete, ~tet)
~zen

5.D.5 Cluster tree 1

Level 1	Level 2	Level 3	Cluster type	Inflection type	Cluster (an. / synth. lex. base)	Examples	Exceptions
L1			basC	reg	~en	loben	gehen
	L2		**homC**	**ei-ie-ie**	**~reiben**	**reiben**	**--**
	L2		basC2	reg / +e(dt)	~den	baden	winden
		L3	**homC**	**i-a-u / +e(dt)**	**~winden**	**winden**	**--**
	L2		**homC**	**i-a-u**	**~lingen**	**gelingen**	**--**
	L2		**homC**	**i-a-u**	**~wingen**	**schwingen**	**--**
	L2		**homC**	**ie-o-o**	**~riechen**	**riechen**	**--**
	L2		**homC**	**ei-i-i**	**~leichen**	**gleichen**	**--**
	L2		homC	reg / +e(mn)	~dmen	widmen	--
	L2		**homC**	**i-o-o**	**~limmen**	**glimmen**	**--**
	L2		homC	reg / +e(mn)	~tmen	atmen	--
	L2		homC	reg / +e(mn)	~bnen	ebnen	--
	L2		homC	reg / +e(mn)	~dnen	ordnen	--
	L2		homC	reg / +e(mn)	~fnen	öffnen	--
	L2		homC	reg / +e(mn)	~gnen	eignen	--
	L2		homC	reg / +e(mn)	~chnen	rechnen	--
	L2		homC	reg / +e(mn)	~cknen	trocknen	--
	L2		homC	reg / +e(mn)	~mnen	ver_vollkommnen	--
	L2		homC	reg / +e(mn)	~pnen	wappnen	--
	L2		basC2	reg / -ge(ieren)	~ieren	studieren	stieren
	L2		basC2	reg / -s(sz)	~sen	grasen	blasen
	L2		basC2	reg / -s(sz)	~ßen	spaßen	fließen
		L3	**homC**	**ie-o-o / -s/+e(sz)/+CC**	**~ließen**	**fließen**	**--**
		L3	**homC**	**ie-o-o / -s/+e(sz)/+CC**	**~rießen**	**sprießen**	**--**
	L2		basC2	reg / +e(dt)	~ten	waten	raten
	L2		homC	reg / -s(sz)	~xen	faxen	--
	L2		basC2	reg / -s(sz)	~zen	pflanzen	sitzen
L1			homC	reg / -e(lr)	~eCn	handeln, zaubern	--

5.D.5 Cluster tree 2 (not recommended)

basC2 ~Cmen (reg / +e(mn)) would require
basC3 ~hmen (reg), homC ~lmen (reg),
basC3 ~mmen (reg), homC ~rmen (reg)

basC2 ~Cnen (reg / +e(mn)) would require
homC ~Vhnen (reg), basC3 ~nnen (reg), homC ~rnen (reg)

Level 1	Level 2	Level 3	Level 4	Cluster type	Inflection type	Cluster (an. / synth. lex. base)	Examples	Exceptions
	L2			basC2	reg / +e(mn)	~Cmen	atmen	kommen
		L3		basC3	reg	~hmen	nach_ahmen	nehmen
		L3		homC	reg	~lmen	qualmen	--
		L3		basC3	reg	~mmen	stammen	glimmen
			L4	**homC**	**i-o-o**	**~limmen**	**glimmen**	**--**
		L3		homC	reg	~rmen	formen	--
	L2			basC2	reg / +e(mn)	~Cnen	ebnen	kennen
		L3		homC	reg	~Vhnen	dehnen	--
		L3		basC3	reg	~nnen	sonnen	nennen
		L3		homC	reg	~rnen	lernen	--

Part of the cluster tree

5.D.6 Part of the lexeme register

Flag	Cluster	Infl. type	Analytic lex. base	Synthet. lex. base	Present 3rd sg	Past tense 1st sg	Past Participle	Comment
s	basC	reg	~en	~en		~te	ge~t	e.g. loben
		reg	säen	säen		säte	gesät	
s		**a-a-a/Aux**	**haben**	**haben**	**hat**	**hatte**	**gehabt**	
s		**a-a-a/Aux**	**inne_haben**	**innehaben**	**hat inne**	**hatte inne**	**innegehabt**	
p		reg	schaben	schaben		schabte	geschabt	
s		reg	handhaben	handhaben		handhabte	gehandhabt	noun Handhabe
s		**a-ä-u-a**	**graben**	**graben**	**gräbt**	**grub**	**gegraben**	
p		reg	traben	traben		trabte	getrabt	
s		**e-i-a-e**	**geben**	**geben**	**gibt**	**gab**	**gegeben**	**imper. gib**
s		**e-o-o**	**heben**	**heben**		**hob**	**gehoben**	
		e-o-o	**auf_heben**	**aufheben**	**hebt auf**	**hob auf**	**aufgehoben**	
s		**ie-o-o**	**schieben**	**schieben**		**schob**	**geschoben**	
p		reg	lieben	lieben		liebte	geliebt	
s		reg	α+stieben	stieben		stiebte	gestiebt	
v		**ie-o-o**	**β+stieben**	**stieben**		**stob**	**gestoben**	
		reg	leben	leben		lebte	gelebt	
s		reg	α+weben	weben		webte	gewebt	
v		**e-o-o**	**β+weben**	**weben**		**wob**	**gewoben**	
p		reg	schweben	schweben		schwebte	geschwebt	
p		reg	einver_leiben	einverleiben		verleibte ein	einverleibt	
s		ei-ie-ie	bleiben	bleiben		blieb	geblieben	
s	**homC**	**ei-ie-ie**	**~reiben**	**~reiben**		**~rieb**	**ge~rieben**	
p		**ei-ie-ie**	**reiben**	**reiben**		**rieb**	**gerieben**	
p		**ei-ie-ie**	**schreiben**	**schreiben**		**schrieb**	**geschrieben**	
p		**ei-ie-ie**	**treiben**	**treiben**		**trieb**	**getrieben**	
p		reg	erben	erben		erbte	geerbt	
s		**e-i-a-o**	**ver_derben**	**verderben**	**verdirbt**	**verdarb**	**verdorben**	**imper. verdirb**
p		reg	gerben	gerben		gerbte	gegerbt	
p		reg	kerben	kerben		kerbte	gekerbt	
s		**e-i-a-o**	**sterben**	**sterben**	**stirbt**	**starb**	**gestorben**	**imper. stirb, subj. past stürbe**
s		**e-i-a-o**	**werben**	**werben**	**wirbt**	**warb**	**geworben**	**imper. wirb**
s		reg	α+schnauben	schnauben		schnaubte	geschnaubt	current use
v		**au-o-o**	**β+schnauben**	**schnauben**		**schnob**	**geschnoben**	**obsolete**
p		reg	rauben	rauben		raubte	geraubt	
s	basC2	reg/+e(dt)	~den	~den	~det	~dete	ge~det	e.g. baden
p		reg/+e(dt)	baden	baden	badet	badete	gebadet	
s		**a-ä-u-a/+e(dt)**	**laden**	**laden**	**lädt**	**lud**	**geladen**	

5.D.7 Result

Es gibt es eine regelmäßige (schwache, produktive) und eine unregelmäßige (starke, nicht produktive) Konjugationsklasse ohne durchgängige Abgrenzungsmöglichkeit über rückläufige Sortierung.

Die reguläre Konjugationsklasse
enthält produktive Subtypen mit regelhaften Anpassungen aufgrund von phonotaktischen, phonetischen und orthographischen Besonderheiten:

***e*-Einschübe bei Verben auf *-den*, *-ten*, *-Cmen* (*C* ungleich *h*, *l*, *m*, *r*; alle homogen), *-Cnen* (nur *-fnen*, *-gnen*, *-chnen* homogen)**

(Mögliche) *e*-Elision im Ind.Prs.1.Sg. und Besonderheiten im Konj.Prs. bei Verben auf *-eln* und *-ern* (beide homogen)

***s*-Elision im Ind.Prs.2.Sg. bei Verben auf *-sen*, *-ssen*, *-ßen*, *-xen* (homogen), *-zen*.**

Ansätze für ausgangsbasierte Analogieschlüsse sind vorhanden, aber nicht zahlreich.
Etwa ein Neuntel (21/190) der unregelmäßigen Verben liegt in homogenen Gruppen ausgangsgleicher mit der Durchschnittsgröße 2 ½ (21/9).

5.E The situation of the English verb

5.E.1 Principal parts, stem distribution

infinite forms		Present infinitive	░	Imperative	
		Gerund Present participle	░	░	2nd sg
		Past participle	▤	░	2nd pl
Present tense	1st sg	░	░	░	1st sg
	2nd sg	░	░	░	2nd sg
	3rd sg	░	░	░	3rd sg
	1st pl	░	░	░	1st pl
	2nd pl	░	░	░	2nd pl
	3rd pl	░	░	░	3rd pl
Past tense	1st sg	▓	▓	▓	1st sg
	2nd sg	▓	▓	▓	2nd sg
	3rd sg	▓	▓	▓	3rd sg
	1st pl	▓	▓	▓	1st pl
	2nd pl	▓	▓	▓	2nd pl
	3rd pl	▓	▓	▓	3rd pl

░ **inflection forms derived from the infinitive**

▓ **inflection forms derived from past tense 1st sg**

▤ **inflection forms derived from past participle**

(Holl / Maroldo / Urban 2007, 94)

5.E.2 Homogeneous clusters

Infl. type	Lex. base	Examples
reg / CC	#CVb	rob
reg / CC	#CCVb	crab
reg / CC	~CVc	frolic
i-i-i / D	**~ild**	**gild, build**
reg / CC	#CCVg	drag
i-u-u / 0	**~ling**	**cling, fling, sling**
reg / +e	~sh	fish (fishes)
reg / CC	#(C)CVk	trek
reg / CC	#CVl	ex_cel, gel, en_rol, ex_tol, an_nul
reg / CC	#CCVl	di_stil
reg / CC	#CVm	rim
reg	~oo	boo (booing, boos, booed)
reg / CC	#CCCVp	strip
ee-e-e / D	**~weep**	**weep, sweep**
ea-o-o / n	**~wear**	**wear, swear**
reg / +e	~ss	kiss (kisses)
reg / +e	~x	fix (fixes)
reg / CC +e	#(C)CVz	quiz (quizzing, quizzes)
reg / +e	~Cz	waltz (waltzes), buzz (buzzes)

homC without examples not mentioned, e.g. #CCCVb

5.E.3 Inhomogeneous clusters

~ch*: hatch, match, watch, fetch* are reg.
teach (taught taught), catch (caught, caught)

etc. (e.g. see cluster tree)

5.E.4 Basic clusters

Cluster information	Basic cluster: morphological property (infl. type)	Basic cluster: alphabetic property (lexical base)	Examples	Exceptions
basC	reg	~C	look	dig
basC	reg	~V	visa, boo	go
basC2; no! caution: ~w, ~x, ~Vy reg bas/homC	reg / CC	#(C)CVC	kid	bid
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#(C)CVd	kid grid	bid clad
basC2 (baking)	reg / -e	~Ce	bake	wake
basC2 (freeing)	reg / e	~ee	free	see
basC2	reg / y(Cie)	~ie	tie	lie
basC2 (toeing)	reg / e	~oe	toe	shoe
basC2 (valuing)	reg / -e	~ue	value	glue (glu(e)ing)
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#CVg	jig	dig
basC2 (matches)	reg / +e	~ch	match	catch, teach
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#CCVm	skim	swim
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#(C)CVn	sun twin	win spin
basC2 (vetoes)	reg / +e	~Co	veto	go
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#CVp, #CCVp	cap stop	kid_nap wor_ship
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#(C)CVr	bar	dif_fer
basC2 (gassing, gasses) monosyllabic basic verbs and their prefixed verbs	reg / CC +e	#(C)CVs	gas	bus
basC2 monosyllabic basic verbs and their prefixed verbs	reg / CC	#(C)CVt	bat flit, glut, smut	get slit, spit split
basC2 (not necessary)	reg	~w	view	know
basC2	reg / ie(Cy)	~Cy	dry	fly
basC2 (not necessary)	reg	~Vy	play	buy

5.E.5 Cluster tree 1

Level 1	Level 2	Level 3	Cluster type	Inflection type	Cluster (an. / synth. lex. base)	Examples	Exceptions
L1			basC	reg	~C	look	dig
	L2		homC	reg / CC	#CVb	rob	--
	L2		homC	reg / CC	#CCVb	crab	--
	L2		homC	reg / CC	#CCCVb	?	--
	L2		homC	reg / CC	~CVc	frolic	--
	L2		**homC**	**i-i-i / D**	**~ild**	**gild, build**	**--**
	L2		basC2	reg / CC	#CVd	kid	bid
	L2		basC2	reg / CC	#CCVd	grid	clad
	L2		homC	reg / CC	#CCCVd	?	_--
	L2		basC2	reg / -e	~Ce	bake	wake
	L2		basC2	reg / e	~ee	free	see
	L2		basC2	reg / y(ie)	~ie	tie	lie
	L2		basC2	reg / e	~oe	toe	shoe
	L2		basC2	reg / -e	~ue	value	glue
	L2		**homC**	**i-u-u / 0**	**~ling**	**cling, fling**	**--**
	L2		basC2	reg / CC	#CVg	jig	dig
	L2		homC	reg / CC	#CCVg	drag	--
	L2		homC	reg / CC	#CCCVg	?	--
	L2		basC2	reg / +e	~ch	watch	teach, catch
	L2		homC	reg / +e	~sh	fish	--
	L2		homC	reg / CC	#CVk	?	--
	L2		homC	reg / CC	#CCVk	trek	--
	L2		homC	reg / CC	#CCCVk	?	--
	L2		homC	reg / CC	#CVl	ex_cel, gel	--
	L2		homC	reg / CC	#CCVl	di_stil	--
	L2		homC	reg / CC	#CCCVl	?	--
	L2		homC	reg / CC	#CVm	rim	--
	L2		basC2	reg / CC	#CCVm	skim	swim
	L2		homC	reg / CC	#CCCVm	?	--
	L2		basC2	reg / CC	#CVn	sun	win
	L2		basC2	reg / CC	#CCVn	twin	spin
	L2		homC	reg / CC	#CCCVn	?	--
	L2		basC2	reg / +e	~Co	veto	go
	L2		homC	reg	~oo	boo	--
	L2		basC2	reg / CC	#CVp	cap	kid_nap
	L2		basC2	reg / CC	#CCVp	stop	wor_ship

Level 1	Level 2	Level 3	Cluster type	Inflection type	Cluster (an. / synth. lex. base)	Examples	Exceptions
	L2		homC	reg / CC	#CCCVp	strip	--
	L2		**homC**	**ee-e-e / D**	**~weep**	**weep, sweep**	**--**
	L2		basC2	reg / CC	#(C)CVr	bar	dif_fer
	L2		homC	reg / CC	#CCVr	?	--
	L2		homC	reg / CC	#CCCVr	?	--
	L2		**homC**	**ea-o-o / n**	**~wear**	**wear, swear**	**--**
	L2		homC	reg / +e	~ss	kiss	--
	L2		basC2	reg / CC +e	#CVs	gas	bus
	L2		homC	reg / CC +e	#CCVs	?	--
	L2		homC	reg / CC +e	#CCCVs	?	--
	L2		basC2	reg / CC	#CVt	bat	get
	L2		basC2	reg / CC	#CCVt	flit, glut, smut	slit, spit
	L2		basC2	reg / CC	#CCCVt	?	split
	L2		homC	reg / +e	~x	fix	--
	L2		basC2	reg / ie(Cy)	~Cy	dry	fly
	L2		homC	reg / +e	~Cz	waltz, buzz	--
	L2		homC	reg / CC +e	#CVz	?	--
	L2		homC	reg / CC +e	#CCVz	quiz	--
	L2		homC	reg / CC +e	#CCCVz	?	--
L1			basC	reg	~V	visa, boo	go

Three-consonant groups at the beginning of words:

scratch, stretch, spring
scl, stl, spl
shrink, shl

5.E.5 Cluster tree 2 (not recommended)

Le-vel 1	Le-vel 2	Le-vel 3	Cluster type	Inflection type	Cluster (an. / synth. lex. base)	Examples	Exceptions
L1			basC	reg	~C	look	dig
	L2		basC2	reg / CC	#CVC	bat	get
	L2		basC2	reg / CC	#CCVC	flit, glut, smut	slit, spit
	L2		basC2	reg / CC	#CCCVC	strip	split
	?		homC	reg / CC	~CVc	frolic	--
	L2		**homC**	**i-i-i / D**	**~ild**	**gild, build**	**--**
	L2		basC2	reg / -e	~Ce	bake	wake
	L2		basC2	reg / e	~ee	free	see
	L2		basC2	reg / y(ie)	~ie	tie	lie
	L2		basC2	reg / e	~oe	toe	shoe
	L2		basC2	reg / -e	~ue	value	glue
	L2		**homC**	**i-u-u / 0**	**~ling**	**cling, fling**	**--**
	L2		basC2	reg / +e	~ch	watch	teach, catch
	L2		homC	reg / +e	~sh	fish	--
	L2		basC2	reg / +e	~Co	veto	go
	L2		homC	reg	~oo	boo	--
	L2		**homC**	**ee-e-e / D**	**~weep**	**weep, sweep**	**--**
	L2		**homC**	**ea-o-o / n**	**~wear**	**wear, swear**	**--**
	L2		homC	reg / +e	~ss	kiss	--
		L3	basC2	reg / CC +e	#CVs	gas	bus
		L3	homC	reg / CC +e	#CCVs	?	--
		L3	homC	reg / CC +e	#CCCVs	?	--
	?		basC2	reg	~w	view	know
	?		homC	reg / +e	~x	fix	--
	L2		basC2	reg / ie(Cy)	~Cy	dry	fly
	?		basC2	reg	~Vy	play	buy
	L2		homC	reg / +e	~Cz	waltz, buzz	--
		L3	homC	reg / CC +e	#CVz	?	--
		L3	homC	reg / CC +e	#CCVz	quiz	--
		L3	homC	reg / CC +e	#CCCVz	?	--
L1			basC	reg	~V	visa, boo	go

5.E.5 Cluster tree 2 (not recommended)

Problems:

#(C)CVC vs. ~CVc:	**not disjoint, no subset, same infl. types**
#(C)CVC vs. ~w:	**not disjoint, no subset, diff. infl. types**
#(C)CVC vs. ~x:	**not disjoint, no subset, diff. infl. types**
#(C)CVC vs. ~Vy:	**not disjoint, no subset, diff. infl. types**
	Is *y* a consonant?
#(C)CVs	**needed anyway (*e*-inserting)**
#(C)CVz	**needed anyway (*e*-inserting)**

5.E.6 Part of the lexeme register

Flag	Cluster	Inflection type	Analytic lex. base	Synthetic lex. base	Past tense	Past participle	Comment
s	basC	reg	~C	~C	~Ced	~Ced	e.g. look
s	basC	reg	~V	~V	~Ved	~Ved	e.g. visa, boo
s	basC2	reg / CC	#CVg	#CVg	CVgged	CVgged	e.g. jig
s	homC	reg / CC	#CCVg	#CCVg	CCVgged	CCVgged	e.g. drag
		reg / CC	drag	drag	dragged	dragged	
s		**i-u-u / 0 / CC**	**dig**	**dig**	**dug**	**dug**	
		reg / CC	jig	jig	jigged	jigged	
		reg / CC	pig	pig	pigged	pigged	
s		reg	1hang	hang	hanged	hanged	kill with a rope
v		**a-u-u / 0**	**2hang**	**hang**	**hung**	**hung**	suspend
		reg	whang	whang	whanged	whanged	
		reg	king	king	kinged	kinged	
s	**homC**	**i-u-u / 0**	**~ling**	**~ling**	**~lung**	**~lung**	**cling, fling, sling**
p		**i-u-u / 0**	**cling**	**cling**	**clung**	**clung**	
p		**i-u-u / 0**	**fling**	**fling**	**flung**	**flung**	
p		**i-u-u / 0**	**sling**	**sling**	**slung**	**slung**	
s		reg	1ring	ring	ringed	ringed	provide with a ring
v		**i-a-u / 0**	**2ring**	**ring**	**rang**	**rung**	sound
s		**i-ou-ou / D**	**bring**	**bring**	**brought**	**brought**	
s		**i-a-u / 0**	**α+spring**	**spring**	**sprang**	**sprung**	BE
v		**i-a-u / 0**	**β+spring**	**spring**	**sprung**	**sprung**	AE
s		**i-u-u / 0**	**string**	**string**	**strung**	**strung**	
s		**i-u-u / 0**	**wring**	**wring**	**wrung**	**wrung**	
s		**i-a-u / 0**	**sing**	**sing**	**sang**	**sung**	
		reg	ting	ting	tinged	tinged	
s		**i-u-u / 0**	**sting**	**sting**	**stung**	**stung**	
		reg	wing	wing	winged	winged	
s		**i-u-u / 0**	**swing**	**swing**	**swung**	**swung**	
		reg	catalog	catalog	cataloged	cataloged	
s		reg / CC	hum_bug	humbug	humbugged	humbugged	
s	basC2	reg / +e	~ch	~ch	~ched	~ched	e.g. reach
		reg / +e	reach	reach	reached	reached	
s		**ea-au-au / D / +e**	**teach**	**teach**	**taught**	**taught**	
		reg / +e	screech	screech	screeched	screeched	
s		reg / +e	α+be_seech	beseech	beseeched	beseeched	
v		**ee-ou-ou / D**	**β+be_seech**	**beseech**	**besought**	**besought**	
		reg / +e	batch	batch	batched	batched	
s		**a-au-au / D / +e**	**catch**	**catch**	**caught**	**caught**	
		reg / +e	scratch	scratch	scratched	scratched	
		reg / +e	watch	watch	watched	watched	

(cf. Holl / Maroldo / Urban 2007, 114-115)

5.E.7 Result

There is a regular (weak, productive) and
an irregular (weak and strong, not productive) conjugation class
(e.g. *learn, learnt, learnt; dig, dug, dug*)
with a very restricted categorisation
via reverse sorting of the present infinitives.

The regular conjugation type
contains productive subtypes with regular adaptations
due to phonotactic, phonetic and orthographic particularities:
consonant doubling (*digging*), *e*-insertion (*wishes*),
***e*-deletion (*aging*), *ie-y* change (*dying*) and *y-ie* change (*studies*).**

There are only few starting points
for analogical reasoning based upon reverse similarity.
Only one twentieth (9/175) of the irregular verbs can be found
in homogeneous clusters of reversely similar verbs
with an average size of 2 (9/4).

6 Benefit of the research results

Better knowledge of the structure of inflectional systems
Compact dictionaries of inflectional systems
Software generator for inflectional forms

Profitable for
- linguists
- language learners
- aphasia patients

7 Future work

Examine further language-part-of-speech combinations

Solve the problems of input and output of regional characters

Design a nicer user interface

Design a simple rule parser to produce all inflectional forms of an arbitrary search lexeme

8 Examples to be presented

9 Literature

Holl, Alfred; Atay, Rumeysa; Çavuş, Ismail: *Rückläufiges Wörterbuch zur türkischen Verbalmorphologie. Aufbereitung mit Datenanalyseverfahren der Informatik (Data Mining).* Nürnberg: Technische Hochschule 2013.

Holl, Alfred: “Inflectional morphology, reverse similarity and data mining – Finding and applying compact and transparent descriptions of verb systems of natural languages”. International Journal for Computer Linguistics Research 2(2011), 2(June), 66-82.

Holl, Alfred; Suljić, Ivan: *Rückläufiges Wörterbuch zur kroatischen Verbalmorphologie. Aufbereitung mit Datenanalyseverfahren der Informatik (Data Mining).* Regensburg: Roderer 2010 [= Studia et exempla linguistica et philologica, Series V: Lexica, Tom. 6].

Holl, Alfred; Zimnik, Gordon: “Data Mining und natürlichsprachliche Verbalmorphologien“. [= Schriftenreihe der Georg-Simon-Ohm-Hochschule Nürnberg 43(2008) 1-54]. Nürnberg: Ohm-Hochschule 2009.

Holl, Alfred; Maroldo, Sara; Urban Reinhard: *The inflectional morphologies of the Swedish noun, the Swedish verb and the English verb. Reverse dictionaries based upon data mining methods*. Växjö 2007 [= Mathematical modelling in physics, engineering and cognitive sciences, vol. 12].

Holl, Alfred; Pavlidis, Stilianos; Urban, Reinhard: *Rückläufiges Wörterbuch zur alt- und neugriechischen Verbalmorphologie. Aufbereitung mit Datenanalyse-Verfahren der Informatik (Data Mining).* Regensburg 2006 [= Studia et exempla linguistica et philologica, Series V: Lexica, Tom. 5].

Holl, Alfred; Behrschmidt, André; Kühn, Alexander: *Rückläufige Register zur russischen und deutschen Verbalmorphologie. Aufbereitung mit Datenanalyse-Verfahren der Informatik (Data Mining).* Regensburg 2004 [= Studia et exempla linguistica et philologica, Series V: Lexica, Tom. 4].

Holl, Alfred: “Datenanalyseverfahren der Informatik (Data Mining) als Grundlage einer didaktischen Darstellung der französischen Verbalmorphologie“. In: Bernhard, Gerald; Kattenbusch, Dieter; Stein, Peter (ed.): *Namen und Wörter. Festschrift Josef Felixberger zum 65. Geburtstag*. Regensburg 2003, 107-119.

Holl, Alfred: “Licht und Schatten von Analogieschlüssen auf der Basis rückläufiger Ähnlichkeit in der Verbalmorphologie romanischer und germanischer Sprachen“. In: Heinemann, Sabine; Bernhard, Gerald; Kattenbusch, Dieter (ed.): *Roma et Romania. Festschrift Gerhard Ernst zum 65. Geburtstag*. Tübingen 2002, 152-167.

Holl, Alfred: “The inflectional morphology of the Swedish verb with respect to reverse order: analogy, pattern verbs and their key forms”. *Arkiv för nordisk filologi* 116 (2001), 193-220.

Holl, Alfred: *Romanische Verbalmorphologie und relationentheoretische mathematische Linguistik. Axiomatisierung und algorithmische Anwendung des klassischen Wort und Paradigma-Modells*. Tübingen 1988 [= Linguistische Arbeiten 216].