Malmberg's interests include child language, bilingualism, language learning and teaching, and he has been engaged in writing or speaking about these topics too. The phonetics chair at Lund was partly motivated by the demands of language teaching. He was also active in starting courses for speech therapists (logopedes). He has published many books on general phonetics and the phonetics of individual languages, and several have been translated into many languages and read by hundreds of thousands of students. The French version of his general phonetics in the series *Que sais-je?* has been printed in 200,000 copies and belongs to the best-sellers in the series. Many Scandinavian language students have also read his popular pocket book *Språket och människan* (Language and Man; 5th ed. 1970) and many remember the nice examples of the speech development of his Finnish foster daughter Sirkka.

Malmberg's most influential survey of linguistics is *Nya vägar inom språkforskningen* (1959; English version *New Trends in Linguistics,* 1964), which opened the eyes of many Ph.D. candidates. Inspired by the mathematical theory of information, a theory which attracted several linguists at the time, he also published *Structural Linguistics and Human Communication* (1963), where he associated linguistics to communication in a new manner. Malmberg has been honoured for his great learning, intellectual capacity and importance in many ways. In 1994 he was bestowed the great price of the Swedish Academy of Letters (Vitterhetsakademin), 'Ann-Kersti och Håkan Swenssons pris', the most prestigeous prize a Swedish humanities scholar can receive. He was a member of many learned societies and academies and also received the French Legion of honour. He was visiting professor at the Sorbonne and became doctor honoris causa there. He has also taught at the universities of Aix-en-Provence, Bloomington (Indiana), Buenos Aires, Ibadan and Sao Paulo. Malmberg was offered the chair in linguistics in Geneva once held by Saussure.

Malmberg has given key note speeches at many conferences and congresses in the world. He had an impressive mastery of the whole field of linguistics. He was a linguist even in the sense of being a polyglot and mastered French, English, German, Spanish and Italian beside his mother tongue Swedish – he preferred to write in French. He was highly appreciated as a teacher, and like other great speakers, he did not want to be interrupted by questions. In his lectures, learning and rhetoric were combined in a way few modern students get a chance to experience.

Despite all his success and tokens of honour, Bertil Malmberg remained a humble person. Friends, colleagues and students all over the world miss him.

Lund University, Dept. of Linguistics
Working Papers 44 (1995), 9–20

Computer implementation of the genitive in Latvian

Inguna Greitāne*

Latvian is one of the Baltic languages, a subclass of the Indo-European family. It is a flectional language and has a rich system of morphology, including seven cases and two numbers. Word order in a Latvian noun phrase can in general be characterized as head final. A genitive in Latvian is used as a tool to bind nouns in a noun phrase and express possession, except with verbal nouns where the genitive may function as an actor or a patient. I have used *Mūsdienu latviešu literārās valodas gramatika* I (1959) and *A grammar of modern Latvian* (Fennell & Gelsen 1980) for reference.

Morphology

Latvian nouns consist of a root and an inflectional ending. The inflectional endings can be determined by the declension of the noun. Latvian nouns are divided into 6 declensions depending on the nominative ending and gender of the noun. (Approximately ten nouns belonging to the second declension have the ending *s* in the nominative and genitive.)

Gender	*Ending*	*Declension*	*Examples*	
masc	s	1	tēv-s	'father'
masc	š	1	ceļ-š	'road'
masc	is	2	kaķ-is	'cat'
masc	s	2	akmen-s	'stone'
masc	us	3	tirg-us	'market'
masc/fem	a	4	mās-a	'sister'
masc/fem	e	5	egl-e	'fire'
fem	s	6	gov-s	'cow'

The genitive is formed by adding the following endings to the root (see Greitāne 1994):

* AI Lab, Institute of Mathematics and Computer Science, University of Latvia.

Declension	*Genitive singular*	*Genitive plural*	*Example*
1	a	u	tēv-s → tēv-a, tēv-u
2	a	u	kaķ-is → kaķ-a, kaķ-u
2	s	u	akmen-s → akmen-s, akmeņ-u
3	us	u	tirg-us → tirg-us, tirg-u
4	as	u	mās-a → mās-as, mās-u
5	es	u	egl-e → egl-es, egļ-u
6	s	u	gov-s → gov-s, govj-u

For second declension nouns in singular and plural, as well as for fifth and sixth declension nouns in plural, the consonant preceding the case ending can be palatalized:

ez-is → ež-a	'hedgehog'
egl-e → egļ-u	'firs'
nakt-s → nakš-u	'nights'

Some of the endings (*u, as, es, s*) are used in the genitive as well as in other cases, e.g. *mājas* – genitive singular, nominative plural. In this case, only an understanding of the whole sentence can help to determine which case is used. An automated parsing system for Latvian always searches for the longest constituent, which gives the correct solution in most cases.

Latvian adjectives have two types of declension: definite and indefinite. The indefinite is used for expressions equivalent to English *a little sister*, and the definite for expressions of the type *the little sister*. The indefinite adjectives have endings similar to the nouns. Masculine adjectives are declined like first declension nouns, and feminine adjectives like fourth declension nouns:

	Gender	*Nom sg*	*Gen sg*	*Gen pl*
Indefinite adjective	masc	s, š	a	u
	fem	a	as	u
Definite adjective	masc	ais	ā	o
	fem	ā	ās	o

The following table shows the nominative and genitive of the adjective *mazs* 'little':

Declension	*Gender*	*Nom sg*	*Gen sg*	*Gen pl*
indefinite	masc	maz-s	maz-a	maz-u
	fem	maz-a	maz-as	maz-u
definite	masc	maz-ais	maz-ā	maz-o
	fem	maz-ā	maz-ās	maz-o

Most Latvian pronouns are declined as nouns of either the first declension (for masculine pronouns) or the fourth (for feminine pronouns) declension. There are some pronouns, for instance *es* 'I', which have special forms for each case.

Functional roles

A genitive noun phrase may have several functions, namely:

(1) an attribute in NP: *māsas galds* 'sister's table'.
(2) a complement in a preposition phrase with the prepositions *aiz* 'behind', *ārpus* 'outside', *augšpus* 'up', *bez* 'without', *dēļ* 'for', *kopš* 'since', *no* 'from', *otrpus* 'on the other side', *pēc* 'after', *pie* 'near', *pirms* 'before', *priekš* 'for', *uz* 'on', *virs* 'up', *zem* 'under' in singular: *aiz mājas* [behind house-GEN] 'behind the house'.
(3) a complement in a postposition phrase with the postpositions *augšā* 'high up', *apakšā* 'down below', *dēļ* 'because of', *iekšā* 'inside', *labad* 'for the sake of', *līdzi* 'along with', *priekšā* 'in front of', *starpā* 'between', *vidū* 'among', *virsū* 'on top': *mājas priekšā* [house-GEN in.front.of] 'in front of the house'
(4) the object in negated possessive expressions: *Man nav naudas* [I haven't money-GEN] 'I have no money'.
(5) the genitive noun phrase may occur without its head: *Tā ir māsas* 'This is sister's.'

Thematic roles

In most cases the genitive noun phrase is possessive (*tēva grāmata* 'father's book') or partitive (*rika maizes* [slice bread-GEN] 'slice of bread'). But it may also have the role of an agent (*zēna dziedāšana* 'boy's singing') or a patient (*zēna nolaupīšana* [boy-GEN kidnapping] 'kidnapping of a boy').

A number of nouns can be used in the genitive to describe another noun. The relation between the genitive word and its head is as between an indeclinable adjective and a noun: *ziemas mētelis* [winter-GEN coat] 'winter coat'.

Forms of genitive noun phrases

The Latvian genitive noun phrase is characterized by three main features:

(1) all the words in the genitive noun phrase are in the genitive;
(2) the genitive noun phrase does not agree with the head noun in case, number or gender;

(3) the genitive noun phrase generally precedes the head noun.

An NP including a genitive NP may be more or less complex. The simplest noun phrase with a genitive consists of a noun in the genitive and a head noun ($np_{case}=n_{gen}+n_{case}$: *māsas galds* 'sister's table'), but both the noun in the genitive and the head noun may be complex NPs.

What is a genitive noun phrase? It can be:

(1) A noun in the genitive ($gnp=n_{gen}$: *māsas* 'sister's').
(2) A personal pronoun in the genitive ($gnp=pronoun_{gen}$: *viņa* 'his').
(3) A noun in the genitive preceded by an adjective phrase (usually with a definite ending) in the same case, number and gender as the noun ($gnp=ap_{gen}+n_{gen}$: *mazās māsas* 'little sister's').
(4) A noun in the genitive preceded by a pronoun in the same case, number and gender as the noun ($gnp=pronoun_{gen}+n_{gen}$: *manas māsas* 'my sister's').
(5) A noun in the genitive preceded by a pronoun and an adjective (with definite ending) in the same case, number and gender as the noun ($gnp=pnoun_{gen}+ap_{gen}+n_{gen}$: *manas mazās māsas* 'my little sister's').

(6) The genitive noun phrase may be recursively preceded by another genitive noun phrase ($gnp=gnp_1+gnp_2$):

māsas galda	'sister's table's'
mazās māsas galda	'little sister's table's'
manas mazās māsas jaunā galda	'my little sister's new table's'

The placement of the genitive noun phrase in the matrix noun phrase

Now we can note the genitive's position in the matrix noun phrase. The following constructions are allowed:

(1) A genitive noun phrase precedes the head noun ($np_{case}=n_{gen}+n_{case}$: *māsas galds* 'sister's table').
(2) A head noun preceded by and determined by an adjective phrase in the same case, number and gender as the head noun and a genitive noun phrase ($np_{case}=ap_{case}+gnp+n_{case}$: *mazais māsas galds* 'little sister's table').
(3) A head noun preceded by a pronoun in the same case, number and gender as the head noun, a genitive noun phrase ($np_{case}=pnoun_{case}+gnp+$ $+n_{case}$: *mans ziemas mētelis* 'my winter's coat').
(4) A head noun preceded by and determined by a pronoun and an adjective phrase (with definite ending) in the same case, number and gender as the head noun and a genitive noun phrase. In this case the genitive noun phrase may be regarded as forming a compound with the head noun ($np_{case}=$ $=pnoun_{case}+ap_{case}+gnp+n_{case}$: *mans jaunais ziemas mētelis* 'my new winter's coat').
(5) A head noun preceded by a genitive noun phrase and an adjective phrase in the same case, number and gender as the head noun ($np_{case}=gnp+ap_{case}+$ $+n_{case}$: *māsas mazais galds* 'sister's little table').

The genitive with numerals

If numerals end in *desmits* 'ten', *simts* 'hundred', *tūkstotis* 'thousand', *miljons* 'million', *miljards* 'thousand millions', then a following noun is in the genitive: *pieci simti kronu* [five hundred crowns-GEN] 'five hundred crowns'.

If there is an indeclinable numeral (*simt* '100', *desmit* '10') then the following noun can be used in the genitive instead of nominative or accusative: *piecdesmit kronu* [fifty crowns-GEN] 'fifty crowns'.

The genitive following the head word

The genitive also follows the head word in quantifier phrases (*daudz* 'many', *maz* 'few', etc): *daudz māsu* [many sister-GEN] 'many sisters'.

A genitive noun which has the property of being divisible can either precede or follow the head noun: *maizes rika* [bread-GEN slice] or *rika maizes* [slice bread-GEN] 'slice of bread'. The genitive has a partitive meaning in these cases.

Variation and ambiguity

As mentioned above, a genitive noun phrase generally precedes the head noun. In most cases an adjective is bound with the noun it precedes: *mazās māsas galds* 'little sister's table'. But a genitive may come between the head noun and its modifying adjective: *mazais māsas galds* 'little sister's table'. Therefore you can build ambiguous noun phrases. For instance, *mazā zēna māsa* [little-GEN/NOM boy-GEN sister-NOM] might be translated as 'a little boy's sister' as well as 'a boy's little sister'.

Implementation of the genitive in PROLOG

The following predicates are based on the predicates used for genitive noun phrases in the MT system SWETRA (see Sigurd 1988, Sigurd et al. 1990, Sigurd 1994). The special predicate *lgnp* is used for the Latvian genitive noun phrase, and the predicate *np2* is used to form a representation of the genitive noun phrase compatible with SWETRA. The Prolog predicates are presented in the order they are implemented in the computer program, beginning with the most complex cases.

The following predicate governs a recursive genitive noun phrase which begins with a noun in the genitive. The predicate *lnoun* analyses the noun, checks whether it is used in genitive and returns its meaning in the variable X. The predicate *lgnp* recursively analyses the remainder of the sentence to determine the rest of the genitive noun phrase and returns its meaning in the variable Y. The predicate *np2* forms a functional representation of the whole genitive noun phrase in the variable Np.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
    lnoun(X,_,_,Number1,g,Sem1),
    lgnp(Y),
    {np2(s(X,[]),Y,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by an adjective phrase in the genitive. The predicate *lap* analyses the adjective phrase, checks whether it is used in the genitive, determines its gender and number and returns its meaning in the variable X. The predicate *lnoun* analyses the noun, checks whether it is used in the same case, number and gender as the adjective phrase and returns its meaning in the variable Y.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
    lap(Art,X,gram(Number,g,Gender)),
     lnoun(Y,agr(_,Gender),_,Number,g,Sem1),
     lgnp(Z),
     {np2(s(X,Y,[]),Z,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by a pronoun in the genitive. The predicate *lpnoun* analyses the pronoun, checks whether it is used in the genitive, determines its number and gender and returns its meaning in the variable X.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
     lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
     lnoun(Y,agr(_,Gender),_,Number,g,Sem),
     lgnp(Z),
     {np2(s(X,Y,[]),Z,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by a pronoun and an adjective phrase in the genitive.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
    lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
    lap(Art,Y,gram(Number,g,Gender)),
    lnoun(Z,agr(_,Gender),_,Number,g,Sem),
    lgnp(W),
    {np2(s(tom(X,Y),Z,[]),W,Np)}.
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by a pronoun and an adjective phrase in the genitive. The expression *s(tom(X,Y),Z,[])* is the functional representation of this genitive noun phrase.

```
lgnp(s(tom(X,Y),Z,[]))-->          /* gnp=pnoungen+apgen+ngen */
    lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
    lap(Art,Y,gram(Number,g,Gender)),
    lnoun(Z,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by a pronoun in the same case, gender and number as the noun. The expression *s(X,Y,[])* is the functional representation of this genitive noun phrase.

```
lgnp(s(X,Y,[]))-->          /* gnp=pronoungen+ngen */
    lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
    lnoun(Y,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by an adjective phrase in the same case, gender and number as the noun.

```
lgnp(s(X,Y,[]))-->          /* gnp=apgen+ngen */
   lap(Art,X,gram(Number,g,Gender)),
   lnoun(Y,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting only of a single noun in genitive. The expression *s(X,[])* is functional representation of this genitive noun phrase.

```
lgnp(s(X,[]))-->     /* gnp=ngen */
   lnoun(X,_,_,Number1,g,Sem1).
```

noun phrase, and the predicate *np2* is used to form a representation of the genitive noun phrase compatible with SWETRA. The Prolog predicates are presented in the order they are implemented in the computer program, beginning with the most complex cases.

The following predicate governs a recursive genitive noun phrase which begins with a noun in the genitive. The predicate *lnoun* analyses the noun, checks whether it is used in genitive and returns its meaning in the variable X. The predicate *lgnp* recursively analyses the remainder of the sentence to determine the rest of the genitive noun phrase and returns its meaning in the variable Y. The predicate *np2* forms a functional representation of the whole genitive noun phrase in the variable Np.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
   lnoun(X,_,_,Number1,g,Sem1),
   lgnp(Y),
   {np2(s(X,[]),Y,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by an adjective phrase in the genitive. The predicate *lap* analyses the adjective phrase, checks whether it is used in the genitive, determines its gender and number and returns its meaning in the variable X. The predicate *lnoun* analyses the noun, checks whether it is used in the same case, number and gender as the adjective phrase and returns its meaning in the variable Y.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
   lap(Art,X,gram(Number,g,Gender)),
   lnoun(Y,agr(_,Gender),_,Number,g,Sem1),
   lgnp(Z),
   {np2(s(X,Y,[]),Z,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by a pronoun in the genitive. The predicate *lpnoun* analyses the pronoun, checks whether it is used in the genitive, determines its number and gender and returns its meaning in the variable X.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
   lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
   lnoun(Y,agr(_,Gender),_,Number,g,Sem),
   lgnp(Z),
   {np2(s(X,Y,[]),Z,Np)}.
```

The following predicate governs a recursive genitive noun phrase which starts with a noun in the genitive preceded by a pronoun and an adjective phrase in the genitive.

```
lgnp(Np)-->          /* gnp=gnp1+gnp2 */
   lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
   lap(Art,Y,gram(Number,g,Gender)),
   lnoun(Z,agr(_,Gender),_,Number,g,Sem),
   lgnp(W),
   {np2(s(tom(X,Y),Z,[]),W,Np)}.
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by a pronoun and an adjective phrase in the genitive. The expression *s(tom(X,Y),Z,[])* is the functional representation of this genitive noun phrase.

```
lgnp(s(tom(X,Y),Z,[]))-->          /* gnp=pnoungen+apgen+ngen */
   lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
   lap(Art,Y,gram(Number,g,Gender)),
   lnoun(Z,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by a pronoun in the same case, gender and number as the noun. The expression *s(X,Y,[])* is the functional representation of this genitive noun phrase.

```
lgnp(s(X,Y,[]))-->          /* gnp=pronoungen+ngen */
   lpnoun(X,agr(_,Gender),_,Number,g,Sem1),
   lnoun(Y,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting of a noun in the genitive preceded by an adjective phrase in the same case, gender and number as the noun.

```
lgnp(s(X,Y,[]))-->          /* gnp=apgen+ngen */
   lap(Art,X,gram(Number,g,Gender)),
   lnoun(Y,agr(_,Gender),_,Number,g,Sem).
```

The following predicate governs a genitive noun phrase consisting only of a single noun in genitive. The expression *s(X,[])* is functional representation of this genitive noun phrase.

```
lgnp(s(X,[]))-->      /* gnp=ngen */
   lnoun(X,_,_,Number1,g,Sem1).
```

The following predicate governs a genitive noun phrase consisting of a single personal pronoun in genitive. The predicate *lpnoun* analyses the pronoun, checks whether it is used in genitive and returns its meaning in the variable X.

```
lgnp(X)-->            /* gnp=pronoun_gen */
    lpnoun(X,agr(_,Gender),_,Number,g,Sem1).
```

The predicate *np2* is used to form a representation of the genitive noun phrase compatible with SWETRA (see Sigurd 1994). The first argument of the predicate is the leftmost genitive noun phrase, the second argument is the rest of the genitive noun phrase, and the third argument is the whole noun phrase.

```
np2(X,s(Y,[]),s(X,Y,[])).
np2(X,s(A,Y,[]),s(X,Y,[])).
np2(X,s(A,Adj,Y,[]),s(X,Adj,Y,[])).
```

To parse a Latvian noun phrase containing numerals the following predicate *lnp* is used. The line *[N],{number(N)}* checks whether the noun phrase begins with a number. The predicate *lnoun* analyses the noun following the number, checks whether it is used in the plural and returns its meaning in the variable X. The next predicate *lnoun* analyses the next noun, checks whether it is used in plural genitive and returns its meaning in the variable Y. The expression *s(N,X,Y)* is the functional representation of this noun phrase.

```
lnp(s(N,X,Y)) -->
   [N],{number(N)},
   lnoun(X,agr(_,Gender),_,pl,Case,Sem),
   lnoun(Y,agr(_,Gender1),_,pl,g,Sem1).
```

Genitive rules in the English module of SWETRA

The following predicates show the generation of English genitive noun phrases from their functional representation in the MT system SWETRA. The predicate *enp* generates the English noun phrase. The predicate *egenp* builds a genitive form with endings *'s* or *'*. The predicate *elexg* finds the translation of the noun through its meaning in the lexicon. The lexical format of SWETRA is generally a matrix called *lex* (with different prefixes according to language) and 10 slots for storing lexical information.

The following predicate *enp* generates a noun phrase consisting of a head noun preceded by the genitive of a proper noun or a noun referring to a person and followed by a post attributive phrase. The predicate *pers_or_prop* checks whether the meaning of the functional representation of the genitive noun phrase is a person or a proper name. The predicate *egenp* generates the genitive from the functional representation of the genitive noun phrase in a variable G. The predicate *elexg* generates the head noun from its functional representation in the variable S. The predicate *epattr* generates a post attributive phrase from its functional representation in the variable T.

```
enp(Agr,s(G,S,T)) -->
          {pers_or_prop(G)},
          egenp(_,G),
          elexg(n,S,_,_,_,_,_,_,_),
          epattr(S,T).
```

The following predicate *enp* generates a noun phrase with the same structure but with an additional adjective phrase. The predicate *eap* generates the adjective phrase from its functional representation in a variable A.

```
enp(Agr,s(G,A,S,T)) --> {pers_or_prop(G)},
          egenp(_,G),
          eap(Agr,A),
          elexg(n,S,_,Agr,_,_,_,_,_),
          epattr(S,T).
```

English uses *of* genitive if the noun is not a proper name or a person in the singular. The following predicate *enp* generates a noun phrase which consists of a head noun followed by a genitive noun phrase with the preposition *of* and a post attributive phrase.

```
enp(Agr,s(G,S,T)) -->
          eart(A,m(def,_)),
          enp(Agr,s(S,[])),
          [of],enp(A2,G),
          epattr(S,T).
```

The following predicate *enp* generates a noun phrase which consists of a head noun preceded by an adjective phrase and followed by a genitive noun phrase with the preposition *of* and a post attributive phrase.

```
enp(Agr,s(G,A,S,T)) -->
        eart(A,m(def,_)),eap(_,A), enp(Agr,s(S,[])),
        [of],enp(A2,G),
        epattr(S,T).
```

The following predicate *egenp* calls the predicate *elexg* which generates a genitive noun phrase from its functional representation in the variable S. In the predicate *elexg* the predicate *enp* generates an English noun phrase in the variable W from its functional representation. The predicate *finscnsynt* converts this noun phrase to a genitive noun phrase in variable W1.

```
egenp(_,S) --> elexg(gen,S,_,_,_,_,_,_,_).

elexg(gen,S,_,_,C1,D1,E1,R1,P1,Y,X) :-
        enp(Ag,S,W,[]),
        finsconsynt(W,W1),
        append(W1,X,Y).
```

The following predicate *finsconsynt* adds the genitive ending to the last word of the genitive noun phrase.

```
finsconsynt(W,Wf) :-
        reverse(W,[S|T]),
        (concat(S_first,s,S), concat(S,"',S1);
        concat(S,"s',S1)),
        reverse(Wf,[S1|T]).
```

Demos

The following demos show functional representations of some Latvian sentences and their translation into English. The functional representation is used as an interlingua in the translation process and as the basis for the generation of the English sentences.

Māsas galds ir netīrs.
[subj(s(s(m(def, _1810), m(sister, pers), []), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _1961)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(m(def, _1810), m(sister, pers), []), m(table, sg), []), [.])]
The sister's table is dirty.

Mazās māsas galds ir netīrs.
[subj(s(s(m(def, _3813), m(little, _3782), m(sister, pers), []), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _4016)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(m(def, _3813), m(little, _3782), m(sister, pers), []), m(table, sg), []), [.])]
The little sister's table is dirty.

Mazais māsas galds ir netīrs.
[subj(s(s(m(def, _6028), m(sister, pers), []), m(little, _5936), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _6179)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(m(def, _6028), m(sister, pers), []), m(little, _5936), m(table, sg), []), [.])]
The sister's little table is dirty.

Māsas mazais galds ir netīrs.
[subj(s(s(m(def, _8116), m(sister, pers), []), m(little, _8141), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _8320)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(m(def, _8116), m(sister, pers), []), m(little, _8141), m(table, sg), []), [.])]
The sister's little table is dirty.

Manas māsas galds ir netīrs.
[subj(s(s(m(my, sg), m(sister, pers), []), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _10429)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(m(my, sg), m(sister, pers), []), m(table, sg), []), [.])]
My sister's table is dirty.

Manas mazās māsas galds ir netīrs.
[subj(s(s(tom(m(my, sg), m(little, _12379)), m(sister, pers), []), m(table, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _12606)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(tom(m(my, sg), m(little, _12379)), m(sister, pers), []), m(table, sg), []), [.])]
My little sister's table is dirty.

Māsas galda stūris ir netīrs.
[subj(s(s(s(m(def, _14554), m(sister, pers), []), m(table, sg), []), m(corner, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _14779)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(s(m(def, _14554), m(sister, pers), []), m(table, sg), []), m(corner, sg), []), [.])]
The corner of the sister's table is dirty.

Mazās māsas galda stūris ir netīrs.
[subj(s(s(s(m(def, _16984), m(little, _16953), m(sister, pers), []), m(table, sg), []), m(corner, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _17262)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(s(m(def, _16984), m(little, _16953), m(sister, pers), []), m(table, sg), []), m(corner, sg), []), [.])]
The corner of the little sister's table is dirty.

Manas mazās māsas jaunā galda stūris ir netīrs.
[subj(s(s(s(tom(m(my, sg), m(little, _19756)), m(sister, pers), []), m(new, _19861), m(table, sg), []), m(corner, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _20116)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(s(s(tom(m(my, sg), m(little, _19756)), m(sister, pers), []), m(new, _19861), m(table, sg), []), m(corner, sg), []), [.])]
The corner of my little sister's new table is dirty.

Mans ziemas mētelis ir netīrs.
[subj(s(m(my, sg), m(wintcoat, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _22491)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(m(my, sg), m(wintcoat, sg), []), [.])]
My winter coat is dirty.

Mans jaunais ziemas mētelis ir netīrs.
[subj(s(tom(m(my, sg), m(new, _2034)), m(wintcoat, sg), [])), pred(m(m(be, pres), [])), obj(m(dirty, _2191)), obj([]), advl([]), advl([]), advl([]), advl([]), co(s(tom(m(my, sg), m(new, _2034)), m(wintcoat, sg), []), [.])]
My new winter coat is dirty.

Acknowledgements

This study was made possible by a Swedish Institute Scholarship for research studies at Lund Universtity. I would like to thank Prof. Bengt Sigurd and Barbara Gawrońska who have contributed very much to this research.

References

Fennell, Trevor G. & Henry Gelsen. 1980. *Grammar of modern Latvian.* The Hague: Mouton Publishers.

Greitāne, Inguna. 1994. 'Algorithms for the inflection of Latvian words'. In *Proceedings of the Latvian Academy of Sciences – Part A*, 32-39. Riga (in Latvian).

Mūsdienu latviešu literārās valodas gramatika. Vol. 1. Riga 1959

Sigurd, Bengt. 1988. 'Translating to and from Swedish by SWETRA – a multilanguage translation system'. *New directions in Machine Translation.* Foris Publications.

Sigurd, Bengt 1994. 'The SWETRA Referent Grammar'. In: Bengt Sigurd (ed.) *SWETRA Grammars for Analysis and Machine Translation.* Lund University press.

Sigurd, Bengt, Mats Eeg-Olofsson, Barbara Gawrońska-Werngren & Per Warter. 1990. *Swetra – a multilanguage translation system* (Praktisk Lingvistik 14). Dept. of Linguistics, Lund.

Lund University, Dept. of Linguistics
Working Papers 44 (1995), 21–40

Swedish applied verbs derived by the prefix *be-*

Claire Gronemeyer

Introduction

This paper examines *be*-prefixation of Swedish verbs and its consequences for the argument structure of the derived verb. The Swedish *be-* will be analyzed as an applicative morpheme which signals an alternation in the grammatical functions of the verb's arguments. The applied verb constructions are especially interesting because they are the result of a complex process which shows the interaction between the morphology and the syntax. To my knowledge the Swedish prefix *be-* has not previously been analyzed as an applicative affix, and this will be shown to be a fruitful analysis of *be-* as well as a theoretically interesting account of applied verb constructions in a Germanic language.

Argument inheritance will be analyzed within the DiSciullo & Williams 1987 morphological theory of word formation, Bierwisch's 1989 lexical theory of derivation, and Baker's 1988a syntactic approach. These theories will be discussed in relation to the topic of *be*-prefixation and the grammatical function changing that typically takes place with applicative verbs. The relationship between the occurrence of the applicative morpheme on a verb and the altered argument structure of the verb is explained by these theories with varying degrees of success. This paper will argue that DiSciullo & Williams cannot deal with complex alternations like the applicatives, and many problems remain unsolved in their framework; Bierwisch's approach clears up most of the problems DiSciullo & Williams have, and yet it still isn't sufficient in analyzing all of the applied verb constructions; while Baker's incorporation theory is suited to this type of phenomenon and does the best job of predicting the function and effect of the Swedish *be-* on the verb's argument structure.

The prefix *be-* entered the Swedish language during the Middle Ages with loan words from Low German (Söderbergh 1967). Although the