BENGT SIGURD

Analyzing Runic Swedish by a computerized grammar

Introduction and abstract

The language on Swedish rune stones written in the 16-type alphabet (futhark) constitutes a very special fragment of the Swedish of its time (about 800–1100). Because of the ritual character of the inscriptions the language is rather standardized. The greatest variation is in proper names. As has long been noted the typical formula is (in English translation): *N raised this stone after M his P*, where *N* and *M* are personal names and *P* is a kinship term. Additional sentences may state that he was a good person or where the person died, e.g. *He fell in Greece.* Depending on the success of the new religion the formula *God help his soul* is sometimes also added (for safety). There are about 3000 rune stones with text of this type.

The purpose of this paper is to demonstrate how a generative phrase structure grammar implemented as a computer program can parse typical Runic sentences and present analyses in terms of subject, predicate, objects and adverbials (functional roles) in addition to word meanings. The grammar used is Swetra (Referent) grammar which has been developed for use in the automatic translation project Swetra, see chapter 1 in Sigurd (1994) and has been used in various automatic translation and text generation projects. Swetra grammar is a variant of the generative phrase structure grammar first presented in Chomsky (1957). Swetra grammar includes a functional representation instead of the typical Chomskyan trees and it uses no deep structure and no transformations. The functional representation can be used as an intermediate (universal) language (interlingua) in automatic translation systems or multilingual text generation system. The Swetra grammar is fairly well elaborated and has good empirical coverage. Most of the grammatical constructions met in texts are accounted for. It has been used for analyzing Swedish, English, German, Danish, Russian, Polish, Latvian and Chinese, in particular in the weather and stockmarket domains.

The paper will at the same time describe a computer program called Rune which is able to interpret typical rune stone texts and translate them into Modern Swedish. In a first step the runes are transliterated into Latin letters. The second step is the parsing of the transliterated text resulting in a functional representation showing the analysis in terms of subject, predicate, objects and adverbials in addition to word meanings according to Swetra grammar. In a third step this functional representation is then used as an intermediate language (interlingua) in the automatic translation into Modern Swedish using an equivalent Modern Swedish grammar. The program may also be run in the other direction translating Modern Swedish into runes. Some inscriptions are offered for demonstration.

The program Rune has been implemented in the programming language LPAProlog. (I am indebted to Johan Dahl for assistance in developing the interface.) It can be transported to other Prolog variants and other computers. The program will be demonstrated on demand at the Department of Linguistics, Lund University. The description in this paper will be given with a view to linguistic readers rather than computer science readers. Some of the grammatical problems are discussed. Knowledge in the formalisms of generative grammar and the programming language Prolog is an advantage for the reader.

The program Rune may possibly be used as an aid in the interpretation of rune stones or as a pedagogical tool for those who want to learn the runes or the language of the rune stones. There is a convenient interface with several windows and menues offering a choice of input language and a choice of operations: transliteration, grammatical analysis, translation (see picture).

The potential of the grammar in analyzing rune inscriptions, will be illustrated in the section called Print-outs. We do not maintain that all rune inscriptions can be analyzed (and understood) by the grammar but a substantial fragment certainly can. In some cases the grammar rightly offers several solutions. In some cases a solution might be wrong.

To my knowledge no formal grammar of Runic Swedish has been written before, but computers have been used for storing texts (see survey by Lena Peterson 1996). Particular runic constructions have been commented on, e.g. by Wessén (1956, 1959) and Jansson (1963, 1976), and the runic language is treated in e.g. Noreen (1904), Krause

(1971) and Antonsen (1975). The runic language is interesting as the oldest Nordic language but also from a general linguistic point of view and the formal grammar presented in this paper pinpoints some of the special features of Runic Swedish.

The rune stones

As the Scandinavian nations were not established in these old days around 1000 and the Vikings spoke more or less the same language (tongue) the language described here could have been called runic Norse, but as most of the stones discussed belong to what is nowadays called Sweden the term Runic Swedish has been preferred. It should be clear, of course, that the Vikings did not only speak in the style used on the rune stones, but the rune inscriptions are almost the only evidence we have and are therefore an important key to Old Swedish. The rune inscriptions written in the 24-type alphabet are generally older and more difficult to interpret. They are not treated in this paper.

Scholars have taken an interest in runes since the 17th century (Bureus), but the modern scientific approach started at the end of the 19th century when the problems of the origin of the runes were discussed in the light of new historical-comparative methods by Wimmer, Bugge and von Friesen. The most elaborate documentation of Swedish rune inscriptions is the series *Sveriges Runinskrifter* published by the Royal Swedish Academy of Letters (Kungl. Vitterhets Historie och Antikvitets Akademien) for each region separately since the beginning of the 20th century. It includes detailed discussions of all linguistic, archeologic and historical matters. Similar series were started in Denmark and Norway. The rune stones of Skåne are treated in the Danish works (by Wimmer and later by Jacobsen & Moltke).

A shorter survey including the most interesting stones is *Runinskrifter i Sverige* (1963, 1976) by the famous rune specialist Sven B. F. Jansson. Lars Rask *Runläseboken* (1990) is a pedagogical presentation of runes and rune inscriptions including lessons in rune reading. The stones mentioned by Rask have been taken as a representatitive set here and some of them are included in the demonstration menue (see Print-outs).

Writing

The runic language treated here is the language written by the 16-type runes. The set of runes allowed in the system Rune is presented in a special window including word space : (colon). Dotted ("stung") runes for e.g. *e* distinguishing it from *i*, *p* distinct from *b* are generally not accepted by the transliteration decoder. The Prolog program can not mix runic writing (font) with Latin writing in the same window.

The grammar will generate and analyze Runic Swedish spelled with transliterated runes. The grammar rules operate on transliterated texts.

There are transliteration rules to decode and encode the runes. The runes are written using the special runic font called *Bryggen*.

The following table presents the runes, the transliteration used and the phonological equivalents generally assumed. As is well-known several phonological distinctions, e.g. consonant voice and minor differences in vowel quality are disregarded by the 16-rune alphabet. The values of some runes are dubious, but this is of little importance for our project. We will not dwell on the writing problems in any detail and give no rune variants.

Runes	transliteration	phonetic values (in IPA)
ᚠ	f	f, v
ᚢ	u	u, v(w), o, y, ö
ᚦ	th (thorn)	þ (voiceless), dh (voiced)
ᚬ	a(n)	(nasal) a/o
ᚱ	r	r
ᚴ	k	k, g
ᚼ	h	h
ᚿ	n	n
ᛁ	i	i, j, e
ᛆ	a	a, ä, e
ᛋ	s	s
ᛏ	t	t, d
ᛒ	b	p, b
ᛘ	m	m
ᛚ	l	l
ᛦ	R	R (palatal r < z)

Only these runes are used in the program and when writing an input text these letters must be picked from a table in a special window. If other rune forms are met on a stone — and there are many variants — they must be identified with and rendered by one of these in order to be processed by our program. Note that the thorn rune ᚦ is transliterated with *th* and R is used for the ᛦ rune.

Nasals (*n*, *m*) were not generally marked (are latent) in the writing resulting in e.g. *sikmutʀ* for *sikmuntʀ* (*Sigmundʀ*), *buati* for *buanti* (*buandi*), *iklat* for *England*. The length of sounds was generally not marked in runic writing (e.g. by double letters).

The spelling of words was, of course, not standardized as in our modern national languages with a long orthographic tradition. In order to cover some of the variation some alternative spellings have been accepted in some lexical rules, e.g. *stain/stin* (stone), *þansi/þasi* (this), *raisþi/risþi/raisti/risti* (raised), *satu/sautu* (set), *þaiʀ/þaʀ* (they), *aftiʀ/iftiʀ/uftiʀ/aft*. Further variants can easily be added to the grammar to increase its coverage. We will not dwell on these problems here.

The Prolog program interprets capital letters as variables and proper names can therefore not be spelled with initial capitals (unless surrounded by ' ').

The computer program and the grammar

The program presented is an experimental variant allowing variable input and operations. The input language may be in 16-type runes, which may then be transliterated by one operation. But one may alternatively input a rune text in Latin letters. Another operation is grammatical analysis of the transliterated text, which results in a functional representation with word meanings. A third operation is translation resulting in Modern Swedish. It is also possible to input modern Swedish, analyze it to get the functional representation and translate it into a runic text. There are two grammars and lexicons in the system. They are bidirectional and can be used both for analysis and generation.

The interface (see picture) includes one window for Latin text, one window for runic text. There are menue windows for the choice of input language and operations. A window with a tablet of 16 runes and space (:) is available when the input language is set at Runic Swedish. A set of inscriptions is available for demonstration.

The syntactic patterns and the lexicon used are based on select examples presented in surveys of rune inscriptions (see references). The program has ad hoc ways of handling some idiosyncracies of the rune inscriptions. A lacking object will be interpreted as 'stone' in sentences equivalent to *Ulf raised after Asmund.* A lacking subject will be interpreted as a pronoun (*he* or *they* depending on the agreement) in sentences equivalent to *died in London, fell in the east.* If required to interpret a rune inscription with unknown words the program will offer solutions where unknown words in nominal positions will be interpreted as proper names.

The grammar rules

The sentence is the basic unit of the grammar. In the Prolog program a sentence is represented as a list of words within square brackets, [ ], with commas between the words.The grammar described here does not generate nor analyze coordinated sentences. How this can be done can be seen in Sigurd 1994. But our grammar covers coordinated noun phrases, which are quite frequent on rune stones (e.g. *Tuki auk Biurn risþu stain þansi*).

The typical runic sentence includes an np which has the functional role of subject (e.g. *Tuki;* in the nominative), a finite verb (e.g. *risþi,* 'raised'; number sg), an np which has the functional role of object (*stain þansi,* 'stone this'; accusative case) and a prepositional phrase which plays the functional role of adverbial (*aftiʀ Tuma;* accusative case).

The first sentence rule below states (when used in analysis) that if we find a noun phrase with the semantic value *N1* in the nominative (*agr(nom,Nu,_)*) followed by a finite verb *V* agreeing with the preceding noun phrase as to number (*agr(_,Nu,_)*) and a following np *N2* in the accusative case (agr(acc,_,_)) and possibly an adverbial phrase this sentence will get the functional analysis shown to the left of the arrow: [*subj(N1),pred(V),obj(N2),advl(A1)*]. The surface case marking has disappeared from this representation and only the semantic values of the words are accounted for. The order of the functional roles within the brackets is arbitrary, but standardized in this way in Swetra grammar.

Following Swetra grammar the mode (declarative, question, imperative, optative) of the sentence is shown in the first slot after the top predicate *s(entence)* and the semantic value of the topicalized (first) constituent of the sentence shown in the second slot (in this case the subject, *N1*). The values of the constituents represented as capital letters are derived by the later phrase and lexical rules.

If read in a generative (synthetic) way the rule states that a *d(eclarative)* sentence with topicalized subject and the functional representation found within square brackets in the third slot can be rendered as an np in the nominativ agreeing with the following finite verb as to number, the finite verb and an np in the accusative, followed by an adverbial phrase.

```
s(d,N1,[subj(N1),pred(V),obj(N2),advl(A1)]) ->
                 np(agr(nom,Nu,_),N1),
                 vfin(agr(_,Nu,_),V),np(agr(acc,_,_),N2),advp(A1).
```

Note how agreement is handled. A value can not be unified with a distinct other value, e.g. *nom* not with *acc*, but a value can be unified with an identical value and an unspecified value: _. The subject noun must be a nominative form which is shown in the agreement complex *agr(nom,Nu,_)*. The number value of the subject (*sg* or *pl*) is given by the variable *Nu* to be used to controle the verb agreement. There are no requirements for a certain gender in the agreement between subject and verb. If *Nu* has the value *sg* in the subject, it must have the same value (or no value _, as for modal verbs) in the verb agreement slot; if the value of *Nu* is *pl* in the subject the verb must have this value too (or no value). This takes care of the variation between e.g. *risti* (*sg*) and *ristu* (*pl*), *fil* (*sg*) and *filu* (*pl*).

A good thing with this formalism (technically, Definite Clause Grammar, DCG) is that it can be run directly as a program in Prolog provided certain phrase and lexical rules are also implemented. If we write to the program:

s(M,T,F,[suin,risti,stain,thansi,aftiR,ulf],[]),
the program will "solve" the variables M, T, F giving,
M=d, (The mode: declarative)
T= m(suin,prop), (The topic)
F=[subj(m(suin,prop),pred(m(raise,past)),obj([m(stone,sg),m(this,_)],
advl([m(after,_),m(ulf,prop))])].
The (transliterated) rune sentence to be analyzed is placed in the fourth slot as a list with commas between the words. If we enter a

functional representation on the other hand, the Prolog program will solve for the variable in the fourth slot and generate a grammatical runic sentence as illustrated by the following call:

```
s(d,m(tuki,prop),[subj(m(tuki,prop)),pred(m(raise,past)),
obj([m(stone,pl),m(these,_)]),advl([]))],S,[])
```

This time the program will solve S and deliver the output: S=[tuki,risti,staina,thisi]. The adverbial was set at [] (empty list).

The program Rune includes equivalent Modern Swedish rules, which use Swedish categories (prefixed with *m*) but the same functional representations and representations of word meanings. This enables the program to translate between Runic and Modern Swedish by writing: s(M,T,F,Runic,[]),ms(M,T,F,ModSwed,[]).

Several sentence rules are needed. The following is the equivalent Modern Swedish sentence rule.

```
ms(d,N1,[subj(N1),pred(V),obj(N2),advl(A1)]) ->
                mnp(agr(nom,_,_),N1),
                mvfin(agr(_,_,_),V),mnp(agr(acc,_,_),N2),madvp(A1).
```

The cases nom and accusative are needed only for pronouns.

Sentence variants with auxiliaries

The following is a variant sentence rule showing how a sentence with the auxiliary *let* plus an infinitive is generated and analyzed. Note that the object is placed before the infinitive in the first rule. This corresponds to the sentence: ***suin lit stain þansi raisa aftiʀ asbiurn faþur sin.*** The word order with the object after the infinitive is caught by the second rule which gives the same functional representation in Swetra grammar. Short comments or examples are given (as in the Prolog program) after the sign %. Still another variant is found on the stone Kyrkstigen, Ed where the object and infinitive verb are placed first, topicalized (*runa rista*). This structure with inverse word order is taken care of in the third rule, where the object (*N2*) and the infinitive (*V2*) are registered as topic. The fourth rule covers the case, where the object only (*N2*) is topicalized.

```
s(d,N1,[subj(N1),pred([V1,V2]),obj(N2),advl(A1)]) ->
    np(agr(nom,Nu,_),N1),aux(agr(_,Nu,_),V1),
    np(agr(ack,_,_),N2),vinf(V2),advp(A1). % Obj before aux

s(d,N1,[subj(N1),pred([V1,V2]),obj(N2),advl(A1)]) ->
    np(agr(nom,Nu,_),N1),aux(agr(_,Nu,_),V1),vinf(V2),
    np(agr(ack,_,_),N2),advp(A1). % Obj after inf

s(d,[N2,V2],[subj(N1),pred([V1,V2]),obj(N2),advl(A1)]) ->
    np(agr(ack,_,_),N2),vinf(V2),aux(agr(_,Nu,_),V1),
    np(agr(nom,Nu,_),N1),advp(A1). % Obj+inf as topic

s(d,N2,[subj(N1),pred([V1,V2]),obj(N2),advl(A1)]) ->
    np(agr(ack,_,_),N2),aux(agr(_,Nu,_),V1),
    np(agr(nom,Nu,_),N1),vinf(V2),advp(A1). % Obj as topic
```

Intransitive sentences

In order to handle typical intransitive sentences the following rules may be added. A typical example is *þaiʀ filu i viking* (they fell in viking campaign). The intransitive finite verb category is denoted by *finvi*. This sentence type generally includes two adverbials — more may be added to all sentence patterns. The second rule shows how sentences with a preposed (topicalized) adverbial is covered (as Rök: *aft uæmod standa runaʀ þaʀ*).

```
s(d,N1,[subj(N1),pred(V),advl(A1),advl(A2)]) ->
        np(agr(nom,Nu,_),N1),finvi(agr(_,Nu,_),V),
        advp(A1),advp(A2). % Subj np as topic

s(d,A1,[subj(N1),pred(V),advl(A1),advl(A2)]) ->
        advp(A1),finvi(agr(_,Nu,_),V), np(agr(nom,Nu,_),N1),
        advp(A2). % Adverbial phrase as topic
```

Predicative sentences

Predicative sentences, i.e. sentences with a copula verb and a predicative make up a special type as they have an adjective or (indefinite) noun phrase as a nominal complement. Typical examples are equivalents of *He was very brave* or *He was (the) leader of (the) gang.* The predicative complement must agree with the subject (and be in the nominative case). The functional representation of such sentences may be a matter of discussion as the predicative is, in fact, similar to an attributive or apposition. In the rule below the value of the predicative occurs with the category label *compl(ement).* A characteristic feature of predicative sentences is the occurence of special verbs (*be*) as predicates and we have chosen to label this category as *cop(ula).* Note how the agreement restriction between the subject *np* and the predicative is rendered. The predicative (*prt*) is simply defined as an np or an adjective phrase (to be defined in the later rules).

```
s(d,N1,[subj(N1),pred(V),compl(P),advl(A1)]) ->
        np(agr(nom,Nu,G),N1),cop(agr(_,Nu,_),V),
        prt(agr(nom,Nu,G),P),advp(A1).
```

```
prt(Agr,N) -> np(Agr,N). % A predicative can consist of an np
prt(Agr,A) -> ap(Agr,A). % A predicative may be an adjective
```

Noun phrases and modifiers

The following are some rules of noun phrases, which have been presumed in the rules above. They specify different types of noun phrases, including the most frequent coordinated noun phrases. The rules generating individual lexical items are presented later. The rules show the importance of agreement in Runic Swedish.

The first rule shows the case where the noun phrase consists of a noun only. The second shows how a pronoun may be an np.

```
np(Agr,N) -> n(Agr,N). % a noun only, e.g. Tuki (nom), Tuka (acc)
np(Agr,N) -> pron(Agr,N). % a pronoun only, e.g. ThaiR (nom)
```

A noun phrase may be more complex and contain a modifier phrase (*mp*) as in the following rules. The modifier phrase may occasionally occur before the noun which is indicated by the second variant. Nominal agreement between the head and the modifier is controlled by the values in the agreement complex *Agr,* the name of the complex variable *agr(Case,Number,Gender).* If there are several word menings they are gathered in a list as is shown by [N,A] in the semantic representation to the left of the arrow.

```
np(Agr,[N,A]) -> n(Agr,N),mp(Agr,A). % buanta kuthan, bruthir sin
np(Agr,[N,A]) -> mp(Agr,A),n(Agr,N). % kuthan buanta
```

Modern Swedish normally only accepts attributives before the head and we will therefore only have one equivalent rule for Modern Swedish, unless we want to reflect the archaic word order of Runic Swedish.

```
mnp(Agr,[N,A]) -> mmp(Agr,A),mn(Agr,N). % god make, sin broder
```

Appositions

Old Norse is famous for its appositions, placed rather freely (see below). The use of the appositions is illustrated by the following rules, where *ap* is the apposition category. Typically, the head of such a noun phrase is a proper noun: *Ulf sun sin*. The second rule takes care of the case where there are two appositions, one apposition before and one after the head. The apposition must agree with its head, which is controlled by the variable *Agr*.

```
np(Agr,[N,A]) -> n(Agr,N),ap(Agr,A). % Tuki bruthir BiarnaR
np(Agr,[N,A,B]) -> ap(Agr,A),n(Agr,N),ap(Agr,B). % bruthur sin
Tuka trak kuthan (acc)
```

The structure of appositions is specified by the following rules. Typically, the noun is a kin or social word, e.g. bruthur (brother; acc), suni (sons; acc), filaka (fellow; acc, sg or pl). The mp may also be a genitive np or possessive pronoun. We will not go into further details here.

```
ap(Agr,[N,A]) -> n(Agr,N),mp(Agr,A). % fathir tuka, bruthir sin
```

The word order in Modern Swedish may be rendered by the rule below, which, in fact, changes the postattributive apposition into a preattributive epitet, e.g. *Tuki bruþir sin* into *sin broder Toke*. The different functions of *sin* and a personal pronoun such as *hans* in Modern Swedish are pinpointed when experimenting with the program. Modern Swedish *Björn reste denna sten efter Toke sin broder* is somewhat strange, if *sin* is to refer to Björn; an alternative is *Toke hans broder*. It is, however, correct to use *sin* if the apposition is changed into a preattributive epitet as in *Björn reste denna sten efter sin broder Toke*. In that position *hans* can hardly be used if the word is to refer to Björn.

```
mnp(Agr,[N,A]) -> map(Agr,A),mn(Agr,N). % Björns broder Tuki
```

Removed appositions

Old Norse is famous for its use of appositions removed from their heads (postponed appositions). We show how this can be handled in our grammar by the following rule which includes a postponed subject apposition after the adverbial phrase. Note that the apposition is required to agree with the subject noun phrase and that the value of the apposition included in the variable *Ap* is inserted after the value *N1* of the subject np in the functional representation just as if it was not removed.

```
s(d,N1,[subj([N1,Ap]),pred(V),obj(N2),advl(A1)]) ->
        np(agr(nom,Nu,_),N1),
        vfin(agr(_,Nu,_),V),np(agr(acc,_,_),N2),advp(A1),
        ap(agr(nom,Nu,_),Ap).
```

This rule takes care of a sentence such as: *Tuki raisti stain þansi aftiʀ Tuma faþir Bjarnaʀ*, where *faþir Bjarnaʀ* is the postponed subject apposition whose content value by this rule is represented in the same way in the functional representation as in: *Tuki faþir Bjarnaʀ raisti stain þansi aftiʀ Tuma*. In the sentence *Tuki raisti stain þansi aftiʀ Tuma faþur sin* the apposition *faþur sin* is in the accusative and must therefore belong to the object *Tumi* (*Tuma*). The grammar will analyze such cases correctly.

The word *sin* gets the meaning representation *m(refl,_)*, and the further interpretation of *sin* is considered a matter of semantic interpretation. Such an interpretation rule could state that a reflexive marker should be identified with (substituted by) the value of the subject of the sentence, often a proper name which identifies the referent.

Coordinated noun phrases

The following are rules for coordinated noun phrases. Such noun phrases coordinated with *auk* are quite common. The second rule takes care of the case where a personal pronoun (*þaiR*) sums up and stresses the coordination. Such cases (*Asbiurn auk Loke þaiR* ...) occur only in subject position. The third rule illustrates how an apposition (in plural) to a coordinated noun phrase is handled. Note that the agreement number of a coordinated np is *pl.*

By using np as the second constituent our rules can also handle multiple coordinations, e.g. *Biurn auk Tuki auk Suin* ... The semantic representation of coordination is a list including the meaning of the nps coordinated and the word *and. Ca* denotes case. The meaning of *þaiR* does not appear in the semantic representation.

```
np(agr(Ca,pl,_),[N1,and,N2]) -> n(agr(Ca,_,_),N1),[auk],
        np(agr(Ca,_,_),N2). % Tuki auk Suin

np(agr(nom,pl,_),[N1,and,N2]) -> n(agr(nom,_,G1),N1),[auk],
        np(agr(nom,_,G2),N2),pron(agr(nom,pl,G3),N3). % Tuki auk
        Suin thaiR

np(agr(Ca,pl,_),[N1,and,N2,E]) -> n(agr(Ca,_,_),N1),[auk],
        np(agr(Ca,_,_),N2),ap(agr(Ca,pl,_),E). % Tuki auk Suin sunir
        BiarnaR (with an apposition)
```

One Swedish equivalent is the following:

```
mnp(agr(Ca,pl,_),[N1,and,N2]) -> mn(agr(Ca,_,_),N1),[och],
        mnp(agr(Ca,_,_),N2). % Toke och Sven
```

Adverbial phrases

The general adverbial expression is a prepositional phrase (*aftiR Tuma*) but some standard adverbial phrases of several words may be regarded as unit idiomatic expressions, e.g. *i uiking* and given a unit semantic representation. The preposition requires the following np to have a certain case (only accusative implemented). Alternatives are given within parentheses and ; between them.

```
advp(m(inviking,_)) -> [i,uiking].
advp(m(intheeast,_)) -> ([i,austr];[a,ustarla]).
advp(m(well,_)) -> [uel].
advp(m(manly,_)) -> [trikila].
advp([P,N]) -> p(Agr,P),np(Agr,N).
advp([]) -> []. % no adverbial
```

Some prepositions

```
p(agr(acc,_,_),m(after,_)) -> ([aftiR];[iftiR]; [aiftiR];[uftiR];[aft]).
p(agr(acc,_,_),m(in,_)) -> [i].
p(agr(acc,_,_),m(for,_)) -> [at].
p(agr(acc,_,_),m(on,_)) -> [a].
```

The following are some equivalent modern Swedish rules

```
madvp([P,N]) -> mp(Agr,P),mnp(Agr,N).
madvp(m(inviking,_)) -> [i,viking].
madvp(m(intheeast,_)) -> ([i,öster).
madvp(m(well,_)) -> [väl].
madvp(m(manly,_)) -> [manligen].

mp(agr(acc,_,_),m(after,_)) -> [efter].
mp(agr(acc,_,_),m(in,_)) -> [i].
mp(agr(acc,_,_),m(for,_)) -> [för].
mp(agr(acc,_,_),m(on,_)) -> [vid].
```

Representation of word meanings

In Swetra grammar all word meanings are written on the form *m(M,Gr)*, where *m* stands for meaning. The lexical meanings (values of the variable *M*) are given in Swetra grammar as "machinese" English-like words and the grammatical meaning as values of the variable *Gr*. For nouns *Gr* takes the values *sg* and *pl*, for verbs *pres, past, imp(erative), conj(unctive), nonf(inite)*. The meaning (universal semantic representation) of *faþir* is thus written: *m(father, sg)* and the meaning of *raisti* is written *m(raise, past)*. The case of the words is not represented in the word meanings of Swetra. Case is considered as a surface phenomenon varying with the syntax of the particular language. Gender is neither represented in the word meanings — it is considered as a feature of the lexicon and morphology of the particular language. Definiteness is regarded as a textual phenomenon which belongs to noun phrases if it is marked in the language (see Gawronska, 1993). Such considerations have made it possible to use Swetra grammar for languages as different as Swedish, English, German, Russian, Latvian and Chinese and perform automatic translation between them with reasonable success.

Adjectives, adverbs, prepositions and conjuntions have no value (underscore) of *Gr*, but are represented by the same formula for technical reasons. The semantic representation of *after* is *m(after,_)*. Only the syntactic category includes information about the rection of the preposition. The following are some lexical items. We have not written any general morphological rules for the restricted domain in focus here. How such rules can be written is described in Sigurd (1994).

Some lexical rules

```
a(agr(nom,sg,m),m(good,_)) –> [kuther]. % nominative form singular
a(agr(acc,sg,m),m(good,_)) –> [kuthan]. % accusative form singular
a(agr(acc,pl,m),m(good,_)) –> [kutha]. % accusative form plural

n(agr(nom,sg,m),m(tuki,prop)) –> [tuki].
n(agr(acc,sg,m),m(tuki,prop)) –> [tuka].
n(agr(nom,sg,m),m(sigmund,prop)) –> ([sikmutr];[sikmuntr).
n(agr(acc,sg,m),m(sigmund,prop)) –> ([sikmut];[sigmunt]).
n(agr(nom,sg,m),m(kunar,prop)) –> [kunar].
n(agr(acc,sg,m),m(kunar,prop)) –> [kunar].

n(agr(nom,sg,m),m(father,sg)) –> [fathir].
n(agr(acc,sg,m),m(father,sg)) –> [fathur].
n(agr(nom,sg,m),m(son,sg)) –> [sun].
n(agr(gen,sg,m),m(son,sg)) –> [sunaR].
n(agr(acc,sg,m),m(son,sg)) –> [sun].
n(agr(nom,pl,m),m(son,pl)) –> [sunir].
n(agr(acc,pl,m),m(son,pl)) –> [suni].
n(agr(nom,sg,m),m(husband,sg)) –> ([buanti];[buati]).
n(agr(acc,sg,m),m(husband,sg)) –> ([buanta];[buata]).
n(agr(nom,sg,m),m(fellow,sg)) –> [filaki].
n(agr(acc,sg,m),m(fellow,sg)) –> [filaka].
n(agr(acc,pl,m),m(fellow,pl)) –> [filaka].
n(agr(acc,sg,m),m(stone,sg)) –> ([stain];[stin]).
```

```
n(agr(acc,pl,m),m(stone,pl))–> ([stina];[staina]).
n(agr(ack,pl,_),m(rune,pl))–> ([runa];[runaR]).
n(agr(nom,sg,n),m(cairn,sg)) –> ([kumbl];[kubl]).
n(agr(ack,sg,n),m(cairn,sg)) –> ([kumbl];[kubl]).
n(agr(nom,pl,n),m(cairn,pl)) –> ([kumbl];[kubl]).
n(agr(ack,pl,n),m(cairn,pl)) –> ([kumbla];[kubla]).
n(agr(ack,pl,m),m(runmark,pl)) –> [merki,siRun].
n(agr(ack,pl,m),m(runmark,pl)) –> [merki].
n(agr(nom,sg,_),m(god,sg)) –> [kuth].
n(agr(ack,sg,_),m(soul,sg)) –> ([sial];[sialu]).

pron(agr(nom,pl,m),m(m,pl)) –> ([thaiR];[thiR]).
pron(agr(nom,sg,m),m(m,sg)) –> ([haa];[ha];[saR]).
pron(agr(acc,_,_),m(refl,_)) –> [sik].

vfin(agr(_,sg,_),m(raise,past)) –> ([risthi];[raisthi];[risti];[raisti]).
vfin(agr(_,pl,_),m(raise,past)) –> ([risthu];[raisthu];[ristu];[raistu]).
vfin(agr(_,sg,_),m(set,past)) –> [sati].
vfin(agr(_,pl,_),m(set,past)) –> [satu].
aux(agr(_,sg,_),m(let,past)) –> [lit].
aux(agr(_,pl,_),m(let,past)) –> [litu].
finvi(agr(_,sg,_),m(fall,past)) –> [fil].
finvi(agr(_,pl,_),m(fall,past)) –> [filu].
finvi(agr(_,sg,_),m(go,past)) –> [fur].
finvi(agr(_,pl,_),m(go,past)) –> [furu].
cop(agr(_,sg,_),m(be,past)) –> ([uaR];[huar];[uas]).
cop(agr(_,pl,_),m(be,past)) –> [uaRu].
cop(agr(_,sg,_),m(be,pres)) –> [iR].
cop(agr(_,pl,_),m(be,pres)) –> [iRu].
```

A rule interpreting unknown words

The following rule may be used when trying to interpret unknown inscriptions. It states that an unknown word (X) in a noun position may be identified as a proper noun with singular agreement. This rule must be used with care otherwise words may be wrongly identified as proper names.

```
n(agr(_,sg,_),m(X,prop)) –> [X].
```

Print-outs of the interpretations of some inscriptions

The inscriptions mentioned are referred to by their usual Swedish names. The first inscriptions are given as runes which are then transliterated. The transliterated inscription (*Inscr*) is then entered into the sentence call:
s(Mode,Topic,Funct,Inscr,[ ])) in order to get the *Mode*, the *Topic* and the functional analysis (*Funct*). For some inscriptions it is possible to get several solutions, but we will not discuss this matter at any lenght in this paper. As mentioned it is possible to use the grammar rules in order to generate as well by inserting a functional representation and ask for a transliterated version and then a runic transcription of it. The numbers _3759, _3339, etc. should be disregarded; they are artifacts of the computer program.

Dagstorp (at the museum Kulturen, Lund)

ᛋᛁᚴᛘᚢᛏᚱ: ᛋᛆᛏᛁ:ᛋᛏᛆᛁᚿ:ᚦᛆᚿᛋᛁ:ᛁᚠᛏᛁᛦ:ᚴᛚᛆᚴᛦ:ᚠᛆᚦᚢᚱ:ᛋᛁᚿ

sikmutr sati stain thansi iftiR klakR fathur sin
subj(m(sigmund, prop)), pred(m(set, past)), obj([m(stone, sg), m(this, _3759)]), advl([m(after, _3339), [m(klak, prop), [m(father, sg), m(refl, _873)]]])
Sigmund satte denna sten efter sin fader Klak

Skårby 1 Marsvinsholm

ᚴᛆᚢᛚᚠᛦ:ᛆᚢᚴ:ᛆᚢᛏᛁᛦ:ᚦᛆᛁᛦ:ᛋᛆᛏᚢ:ᛋᛏᛆᛁᚿ:ᚦᛆᚿᛋᛁ:ᛆᚠᛏᛁᛦ:ᛏᚢᛘᛆ:
ᛒᚱᚢᚦᚢᚱ:ᛋᛁᚿ

kaulfR auk autiR thaiR satu stain thansi aftiR tuma bruthur sin
subj([m(kaulf, prop), and, m(autiR, prop)]), pred(m(set, past)), obj([m(stone, sg), m(this, _3726)]), advl([m(after, _3306), [m(tumi, prop), [m(brother, sg), m(refl, _7161)]]])
Kaulv och Auter satte denna sten efter sin broder Tume

Vallkärra

tufi raisthi staina thisi aftiR kamal buanta sin auk asar sun hans
subj(m(tufi, prop)), pred(m(raise, past)), obj([m(stone, pl), m(these,

_1122)]), advl([m(after, _723), [[m(gamal, prop), [m(husband, sg), m(refl, _4926)]], and, [m(asar, prop), [m(son, sg), m(ma, sg)]]]])
Tove reste dessa stenar efter sin husbonde Gammal och hans son Assar

This interpretation assumes that Gammal and Assar are both objects of the preposition *after*. It has, however, been maintained (see e.g. Wimmer) that the correct interpretation should be Tove reste dessa stenar efter sin husbonde Gammal tillsammans med hans son Assar meaning that Assar helped Tove. It is clear from the verb form *raisþi* that the subject is not plural and the phrase *auk asar sun hans* can hardly be a removed second part of a coordinated noun phrase. If we want to get the interpretation where Assar helped Tove we may take *auk asar sun hans* to be a kind of adverbial phrase equivalent to *even Assar his son* or *together with Assar his son*. One may perhaps also think of the phrase as some kind of removed apposition. We have not implemented these other solutions.

Täby

iarlabaki lit raisa stain thisa at sik kuikuan
subj(m(iarlabaki, prop)), pred([m(let, past), m(raise, inf)]), obj([m(stone, sg), m(this, _1890)]), advl([m(for, _1341), [himself alive]])
Jarlabanke lät resa denna sten åt sig själv i livet

We have regarded *sik kuikuan* as a kind of unit noun phrase.

Hunnestad 1

asbiurn auk tumi thaiR satu stain thansi aftiR rui auk laikfrut suni kuna hantaR
subj([m(asbiurn, prop), and, m(tumi, prop)]), pred(m(set, past)), obj([m(stone, sg), m(this, _1086)]), advl([m(after, _606), [m(rui, prop), and, m(laikfrud, prop), [m(son, pl), [m(kuna, prop), m(hanta, prop)]]]]])
Asbiörn och Tume satte denna sten efter Gunne Hands söner Roi och Lekfrod

Skivarp

tumi raisthi stain thansi aftiR rua filaka sin
subj(m(tumi, prop)), pred(m(raise, past)), obj([m(stone, sg), m(this, _4320)]), advl([m(after, _3900), [m(rui, prop), [m(fellow, sg), m(refl, _1260)]]])
Tume reste denna sten efter sin kamrat Roi

Gårdstånga 1

austi auk kunar raisthu staina thisi aftiR knut auk biurn filaka sina
subj([m(austi, prop), and, m(kunar, prop)]), pred(m(raise, past)), obj([m(stone, pl), m(these, _1320)]), advl([m(after, _921), [m(knut, prop), and, m(biurn, prop), [m(fellow, pl), m(refl, _1251)]]])
Austi och Gunnar reste dessa stenar efter sina kamrater Knut och Björn

Kyrkstigen 1

runa rista lit rahnualtr
subj(m(ragnuald, prop)), pred([m(let, past), m(carve, inf)]), obj(m(rune, pl)) advl([])
Runor rista lät Ragnvald

This text is interesting as it shows how the main infinitive verb with its object can be preposed (topicalized). This feature is kept in the Swedish translation.

Kyrkstigen 2

huar a kriklanti
subj(m(m, sg)), pred(m(be, past)), advl([]), advl([m(in, _921), m(greece, prop)])
Han var i Grekland

This inscription lacks a subject, but the program assumes that it is the equivalent of *he*, represented by *m(m, sg)*.

Kyrkstigen 3

uas lis forunki
subj(m(m, sg)), pred(m(be, past)), compl([m(chief, sg), m(gang, sg)]), advl([])
Han var gängs ledare

This inscription also lacks a subject, and the program assumes that it is *m(m, sg)*. The Swedish translation should include a definite article (*gängets*; An English translation would be: *He was the leader of the gang*), but this program does not insert articles.

Hällestad

saR flu aiki at ubsalum
subj(m(m, sg)), pred(m(flee, past)), advl(m(nix, _3429)), advl([m(at,_), m(uppsala, prop)])
Han flydde ej vid Uppsala

We note that this inscription includes the pronoun *saʀ*.

The following example shows how the common formula equivalent to *God hel his soul* is interpreted.

s(M, T, F, [kuth, hialbi, sial, hans], [])
No.1 : M = opt, T = m(god, prop),
F = [subj(m(god, prop)), pred(m(help, conj)), obj([m(soul, sg), m(ma, sg)])]

English translation: God help his soul

The following examples show how the program recognizes the difference between an object apposition (*suni biarnaʀ*) and a subject apposition (*sunir biarnaʀ*).

1) s(M, T, F, [tuki, auk, austi, thaiR, raisthu, stain, thansi, aftiR, tuma, auk, asur, suni, biarnaR], [])
No.1 : M = d, T = [m(tuki, prop), and, m(austi, prop)],
F = [subj([m(tuki, prop), and, m(austi, prop)]), pred(m(raise, past)), obj([m(stone, sg), m(this, _80577)]), advl([m(after, _80505), [m(tumi, prop), and, m(asur, prop), [m(son, pl), m(biorn, prop)]]])]

```
2) s(M, T, F, [tuki, auk, austi, thaiR, risthu, stain, thansi, aftiR, tuma,
auk, asur, sunir, biarnaR], [])
No.1 : M = d, T = [m(tuki, prop), and, m(austi, prop)],
F = [subj([[m(tuki, prop), and, m(austi, prop)], [m(son, pl), m(biorn,
prop)]]), pred(m(raise, past)), obj([m(stone, sg), m(this, _91362)]),
advl([m(after, _91290), [m(tumi, prop), and, m(asur, prop)]])]
```

The following print-out shows how the ambiguous apposition *sun biarnaR* is analyzed either as belonging to the subject or the object.

```
s(M, T, F, [tuki, raisthi, stain, thansi, aftiR, tuma, sun, biarnaR], [])
No.1 : M = d, T = m(tuki, prop),
F = [subj(m(tuki, prop)), pred(m(raise, past)), obj([m(stone, sg),
m(this, _6129)]), advl([m(after, _6057), [m(tumi, prop), [m(son, sg),
m(bjorn, prop)]]])]

No.2 : M = d, T = m(tuki, prop),
F = [subj([m(tuki, prop), [m(son, sg), m(bjorn, prop)]]),
pred(m(raise, past)), obj([m(stone, sg), m(this, _76086)]),
advl([m(after, _76014), m(tumi, prop)])]
```

The following is an example with several unknown (made-up) names showing how the special lexical rule interprets unknown words as proper names

```
s(M, T, F, [kulir, auk, ausi, satu, stain, thansi, aftiR, tilt, bruthur, sin],
[])
No.1 : M=d,T = [m(kulir, prop), and, m(ausi, prop)],
F = [subj([m(kulir, prop), and, m(ausi, prop)]), pred(m(set, past)),
obj([m(stone, sg), m(this, _58251)]), advl([m(after, _58179), [m(tilt,
prop), [m(brother, sg), m(refl, _54777)]]])]
```

Stenkvistastenen (Södermanland), with coordination of three names

```
(M, T, F, [helki, auk, fraykaiR, auk, thorkautr, raistu, merki, siRun, at,
thiuthmunt, fathur, sin], [])
No.1 : M = d, T = [m(helki, prop), and, [m(fraykaiR, prop), and,
m(thorkautr, prop)]],
F = [subj([m(helki, prop), and, [m(fraykaiR, prop), and, m(thorkautr,
prop)]]), pred(m(raise, past)), obj(m(runmark, pl)), advl([m(for,
_32778), [m(thiuthmunt, prop), [m(father, sg), m(refl, _43713)]]])]
```

English translation: Helge and Fraykair and Thorkautr raised these rune marks for Thiutmund their father.

This example shows the correct interpretation of a coordination of three noun phrases. The example also shows that the reflexive pronoun *sin* must be rendered by *their* in English. Obviously, the number of the subject must be taken into account. We will not discuss how this can be built into the program.

Literature

Antonsen, E. H. 1975. *A Concise Grammar of the Older Runic Inscriptions.* Tübingen. (Sprachstrukturen, Reihe A. Historische Sprachstrukturen 3.)

Chomsky, Noam. 1957. *Syntactic Structures.* The Hague: Mouton.

Clocksin, W. & Mellish, C. S. 1981. *Programming in Prolog.* Berlin: Springer.

Erikson, O. & J. P. Strid. 1991. *Runstenar.* Malmö: Edition Erikson.

Gawronska, B. 1993. *An MT oriented Model of Aspect and Article Semantics.* Lund: Lund University press.

Jacobsen, L. & E. Moltke. 1941–1942. *Danmarks Runeindskrifter,* 3 volumes. København.

Jansson, B. F. 1963; 2:a uppl. 1976. *Runinskrifter i Sverige.* Stockholm: AWE/GEBERS.

Krause, W. 1971. *Die Sprache der urnordischen Runeninschriften.* Heidelberg.

Lindquist, Ivar. 1973. *Poesin i inskrifterna på Hällestad-stenarna.* Lund: Studentlitteratur.

Noreen, A. 1904. *Altschwedische Grammatik.* Halle.

Peterson, L. 1996. "Runologi. Försök till ett aktuellt signalement." *Saga och Sed. Kungl. Gustav Adolfs Akademiens årsbok 1995.* Uppsala: Textgruppen i Uppsala, pp. 39–54.

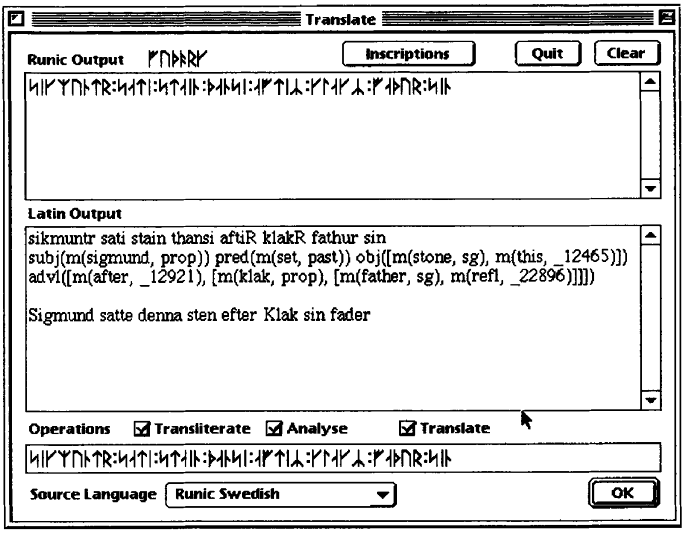
Rask Lars. 1990. *Runläseboken.* Stockholm: Utbildningsförlaget Brevskolan.

Sigurd, Bengt. 1994. *Computerized grammars for analysis and machine translation.* Lund: Lund University press.

Wessén, Elias. 1956. *Svensk språkhistoria III.* Grundlinjer till en historisk syntax. Stockholm: Almqvist & Wiksell.

Wessén, Elias. 1955 (5:th ed.) *Svensk språkhistoria I.* Ljudlära och böjningslära. Stockholm: Almqvist & Wiksell.

Wimmer, L. 1894–1908. *De danske Runemindesmærker,* 4 volumes. København.



ᛋᛁᚴᛘᚢᚿᛏᚱ:ᛋᛆᛏᛁ:ᛋᛏᛆᛁᚿ:ᚦᛆᚿᛋᛁ:ᛆᚠᛏᛁᛦ:ᚴᛚᛆᚴᛦ:ᚠᛆᚦᚢᚱ:ᛋᛁᚿ

```
sikmuntr sati stain thansi aftiR klakR fathur sin
subj(m(sigmund, prop)) pred(m(set, past)) obj([m(stone, sg), m(this, _12465)])
advl([m(after, _12921), [m(klak, prop), [m(father, sg), m(refl, _22896)]]])

Sigmund satte denna sten efter Klak sin fader
```

Operations ☑ Transliterate ☑ Analyse ☑ Translate

ᛋᛁᚴᛘᚢᚿᛏᚱ:ᛋᛆᛏᛁ:ᛋᛏᛆᛁᚿ:ᚦᛆᚿᛋᛁ:ᛆᚠᛏᛁᛦ:ᚴᛚᛆᚴᛦ:ᚠᛆᚦᚢᚱ:ᛋᛁᚿ

Source Language Runic Swedish ▾

The main window of the computer program Rune showing transliteration, analysis and translation to modern Swedish of the Dagstorp inscription.