Magnusson, E. & K. Nauclér. 1991 b. 'On the development of reading in good and poor readers'. *International Journal of Applied Linguistics* 1, 174-185.

Magnusson, E. & K. Nauclér. 1993. 'The development of linguistic awareness in language-disordered children'. *First Language* 13, 93-11.

Rubin, H., M. Kantor & J. Macnab. 1990. 'Grammatical awareness in the spoken and written language of language-disabled children'. *Canadian Journal of Psychology* 44 (4), 483-500.

Ryan, E.B. & G.W. Ledger. 1979. 'Grammaticality judgments, sentence repetitions and sentence corrections of children learning to read'. *International Journal of Psycholinguistics* 6, 23-40.

Ryan, E.B. & G.W. Ledger. 1984. 'Learning to attend to sentence structure: Links between metalinguistic development and reading'. In J. Downing & R. Valtin (eds.), *Language awareness and learning to read*, 149-172. New York, NY: Springer-Verlag.

Appendix

Acceptable and unacceptable sentences used in identification and correction tasks.

Acceptable sentences	Unacceptable sentences	Corrected sentences
Agreement		
Vår bil är ny.	*Min katten är snällt.	Min katt är snäll.
	*Våra huset är liten.	Vårt hus är litet.
Word-order		
Där ligger katten.	*Här vi bor.	Här bor vi.
	*Nu slut glassen.	Nu är glassen slut.
Negation		
Flickan vill inte leka.	*Katten nej sova.	Katten vill inte sova.
		Katten sover inte.
	*Kalle nej äta.	Kalle vill inte äta.
		Kalle äter inte.
Wh-questions		
Vad ska katten äta?	*Katten göra?	Vad gör katten?
		Vad ska katten göra?
	*Pojken heter?	Vad heter pojken?
		Vad ska pojken heta?

Lund University, Dept. of Linguistics
Working Papers 42 (1994), 141-158

Automatic Translation of Knitting Instructions – KNITTRA

Anders Nordner

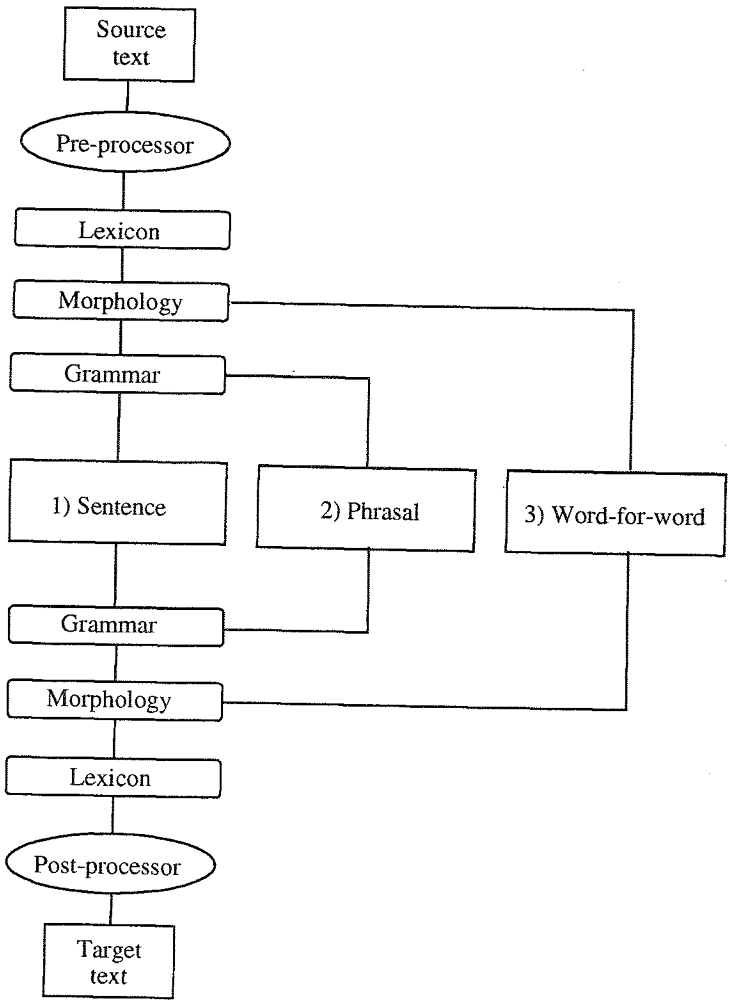
Introduction

The purpose of this paper is to present the results from the work with automatic translation of knitting instructions, KNITTRA – a part of the SWETRA project. SWETRA has earlier treated weather and stock-market reports (cf. Sigurd et al. 1990, 1992).

The languages treated in KNITTRA are Danish and Swedish. Swedish is the target language and the system is not bi-directional. KNITTRA deals with a domain which is as normalised as weather reports, but the vocabulary and the phraseology are different. The sentence types are mainly imperatives and passives apart from many special abbreviated and elliptic sentences.

The translation is carried out by a source and a target language module, each consisting of a lexicon, morphological rules which expand the lexicon, and grammar rules. The source module delivers a syntactic-semantic representation which is used as an interlingua and is fed into the target language module.

KNITTRA first tries to do a complete syntactic analysis of the sentence in order to make a sentence-by-sentence translation. If this does not work, KNITTRA uses a phrasal approach. Finally, a word-for-word translation is also implemented, which is utilised if neither the sentence-by-sentence nor the phrasal approach succeed. A time variable can be set, to three minutes for instance, allowing the program to try sentence-by-sentence translation for that amount of time before the phrasal translation is started. If the time allowed for the phrasal translation is not enough, a word-for-word translation is started. The word-for-word method will always produce a translated sentence or phrase. The quality of the translation depends on the method used. The best results emerge from the sentence-by-sentence translation. The phrasal approach works well with the languages treated, since the syntactic structures are similar. The diagram below shows a brief overview of the program.



The knitting domain

Instructions form a type of text not dealt with in earlier implementations of SWETRA. The following text is a typical part of some knitting instructions:

> Strikkefasthed: 17 m og 21 p på p 5 i glatstrik = 10 x 10 cm. Ribkant: * 1 r, 2 vr *. Ryg og forstykke: Jakken strikkes frem og tilbage på rundp og deles ved ærmegabet. Slå 180 - 189 - 195 m op på rundp 3.5 med rødt, og strik 5 cm ribkant, skift efter 2 p til koral. Skift til rundp 5, samtidig med at der tages 24 - 27 - 33 m ud jævnt fordelt på 1. p = 204 - 216 - 228 m. Husk strikkefastheden! Strik mønster således (mønster 1 skal måle 18 cm i højden, hvis ikke, så afpas efter det glatstrikkede): 5 - 6 - 1 cm glat , mønster 1 , 7 - 9 - 9 cm glat. Str small og medium: Mønster 2, mønster 3, strik færdig med glat. Str large: Strik færdig med mønster 1. Samtidig strikkes: Når arb måler 40 - 45 - 50 cm, deles arb således: 50 - 53 - 56 m forstykke, luk 2 m af, 100 - 106 - 112 m ryg, luk 2 m af, 50 - 53 - 56 m forstykke. Hver del strikkes færdig for sig. Ryg: = 100 - 106 - 112 m. Fortsæt mønsteret. Luk til ærmegab på hver 2. p for 2 m x 2, 1 m x 2 - 1 - 2 = 88 - 96 - 100 m.

Knitting instructions are characterised by sentences in the imperative and passive and also by the use of abbreviations. The vocabulary is very limited – approximately 500 words are found in the source material. The result of an automatic translation of this text is presented in the last section of this paper.

There are references to pictures or diagrams (*) and also to yarn brands in the instructions. If these symbols and names are stored as proper names in the lexicon or if they are interpreted as proper names in the pre-processor (described later in this paper), sentences containing them can be parsed as any other sentence.

Many special symbols are found in the instructions, e.g. the equals sign (=) which can be treated as a predicate. Thus introduced in the lexicon, sentences containing these symbols can also be parsed.

Another characteristic feature is the measurement phrases. These phrases are solved using lexical entries with variable constituents.

The format of the grammar rules, the morphological rules, and the lexicon in other SWETRA projects (STOCKTRA, WEATHRA) has been retained, but specialised texts like these require additional grammar rules and new types of lexical entries.

Some differences between Danish and Swedish

Danish and Swedish have much in common, such as the syntactic structures which are very much the same. However, there are differences, for example in the spelling and in the meaning of many words that look very similar. If one knows Danish, one can generally understand written Swedish and vice versa. This section gives a brief survey of grammatical differences which were important for the design of KNITTRA. For further details on this subject, see Nielsen & Lindegårdh Hjorth 1959 or Kristensen & Brink 1986.

Nouns

Both languages use the same grammatical genders (neuter and common), but words do not always belong to the same class in both languages even though the spelling and meaning are the same, e.g. *et gardin* but *en gardin* in Swedish ('a curtain') and *en krig* but *ett krig* in Swedish ('a war').

In Danish, when using a definite article, the following noun should not be inflected, whereas the construction in Swedish generally requires the noun to be in the definite form after the definite article, e.g. *den gamle præst* and *den gamle prästen* in Swedish ('the old priest'). In this example, the Danish definite article is *den* and in Swedish definiteness is marked by *den* and the ending *-en* in *präst-en*.

Verbs

The non-finite verb form preterite participle in Danish corresponds to two different forms in Swedish, the supine and the past participle, e.g. *en bøjet mand* and *han har bøjet sig* but *en böjd man* and *han har böjt sig* in Swedish ('a bent man' and 'he has bent'). The preterite participle in Danish is inflected only according to the subject's number, whereas in Swedish, the inflection is dependent on both number and gender, e.g. *hunden er bundet* and *dyret er bundet* but *hunden är bunden* and *djuret är bundet* in Swedish ('the dog is tied' and 'the animal is tied'). Danish *hundene er bundne* and *dyrene er bundne* is equal to the Swedish *hundarna är bundna* and *djuren är bundna* ('the dogs are tied' and 'the animals are tied').

Both languages use either an auxiliary (*blive* and *bliva*) or an ending (*-s* in both languages) to express the passive. In the infinitive and the present and the preterite tenses, the *-s* form is used in both languages, e.g. *at kastes, kastes, kastedes* and *att kastas, kastas, kastades* in Swedish ('to be thrown', 'is thrown', 'was thrown'). When forming the passive from the preterite participle in Danish, the auxiliary *blive* must, however, be used, whereas in Swedish there is the possibility to use either *-s* or the auxiliary *bliva*. Danish *er blevet kastet* corresponds to either *har kastats, har blivit kastat, har blivit kastad*, or *har blivit kastade* in Swedish.

In Danish, two auxiliaries may be used for the perfect past, either *være* or *have* ('be' and 'have'). The latter is used for transitive and intransitive imperfective verbs, whereas *være* is used for intransitive perfective verbs. In Swedish, the verb used is always *hava* (corresponding to the Danish *have*). There is one exception in Danish though: in association with the verb *blive*, the only valid auxiliary is *være*.

Some verbs (both Danish and Swedish) have a particle connected to them. The location of the particle is different in the two languages. Generally the particle is located after the object in Danish, whereas it is located before the object in Swedish. The following extract shows that KNITTRA can translate such sentences. Note that the particle is not present in the functional representation. The English translation is 'Cast on 12 stitches':

```
Slå 12 masker op.
[pred(m(cast_on,imp)),obj(s(m(indef,_42E0),m(s(n_stitches,12),pl),[])),o
bj([]),advl([]),advl([]),advl([]),co(_37B8,[.])]
Lägg upp 12 maskor.
Used time: 0 m 0,49 s.
```

The lexicon of KNITTRA

The lexicon of KNITTRA is constructed in approximately the same way as in WEATHRA and STOCKTRA. In the survey below, the slots in the lexical entries are explained for different word classes. The first argument is always the word class, and the second argument is used for a word meaning representation, *m(Meaning,G)*, where *G* is a grammatical meaning, e.g. *sg* or *pl* for nouns, and *pres* for verbs in present tense, *past* for verbs in the past tense, and *imp* for verbs in the imperative. For non-finite verbs, the value is *nonf* and the non-finite form is given in the fourth slot, e.g. *inf* and *part* (infinitive and past participle). The *m(Meaning,G)* part is utilised in the morphological rules (see below). The other arguments are specific to each word class. Words which must agree have the agreement information in the fourth slot. The information in the last slots are reserved for inflection.

The lexical form, e.g. a word or a phrase, is given after the arrow in the DCG (Definite Clause Grammar) format. The following are extracts from the lexicon showing examples of words of different word classes. The predicate *dlex* is interpreted as Danish lexicon.

```
dlex(a,m(ready,_),_,agr(sg,n,indef),_,_,_,_,_) --> ['færdigt'].
dlex(a,m(ready,_),_,agr(pl,_,_),_,_,_,_,_) --> ['færdige'].
dlex(a,m(ready,_),_,agr(sg,r,indef),_,_,_,_,_) --> ['færdig'].
dlex(a,m(fine,compa),_,agr(_,_,_),_,_,_,_,r) --> ['finere'].
dlex(a,m(firm,_),_,agr(sg,_,indef),_,_,_,_,r) --> ['fast'].
```

```
dlex(gradv,m(more,_),_,_,_,_,_,_) --> ['mere'].
dlex(adv,m(important,_),_,_,_,_,_,_) --> ['vigtigt'].
dlex(adv,m(such,_),_,_,_,_,_,_) --> ['sådan'].
dlex(art,m(indef,sg),indef,agr(sg,r,indef),_,_,_,_) --> [en].
dlex(art,m(indef,sg),indef,agr(sg,n,indef),_,_,_,_) --> [et].
dlex(n,m(sleeve,sg),indef,agr(sg,n,indef),_,_,e,r,r2) --> ['ærme'].
dlex(n,m(armhole,sg),indef,agr(sg,n,indef),_,_,c,r,r1) -->
    ['ærmegab'].
dlex(n,m(arm_dome,sg),def,agr(sg,r,def),_,_,_,i,i) -->
    ['ærmekuplen'].
dlex(n,m(arm_dome,sg),indef,agr(sg,r,indef),_,_,_,i,i) -->
    ['ærmekuppel'].
dlex(n,m(embroidery,sg),indef,agr(sg,n,indef),_,_,i,r,r3) -->
    [broderi].
dlex(p,m(to,_),_,_,_,_,_,_) --> ['til'].
dlex(p,m(on,_),_,_,_,_,_,_) --> ['på'].
dlex(v,m(passive,nonf),aux,part,_,part,_,i,[]) --> [blevet].
dlex(v,m(describe,nonf),vt,part,_,_,_,_,[]) --> [beskrevet].
dlex(v,m(use,nonf),vt,inf,_,_,_,r,[]) --> [bruge].
dlex(v,m(cast_on,imp),vt,fin,_,_,_,i,[op,emph]) --> ['slå'].
dlex(v,m(hit,imp),vt,fin,_,_,_,i,[]) --> ['slå'].
```

Multi-word lexical entries

Certain frequently occurring phrases are stored directly in the lexicon and are treated as multiple word lexical entries (cf. Sigurd 1993). In this way it is possible to extract a larger chunk (more than a single word) of source text and replace it with the corresponding target text. This increases the quality drastically and decreases the time used for translation. Instead of looking for all the constituent words in a phrase, one can look for larger chunks of text and translate them directly. This technique is also used by human translators (cf. Karlgren 1988). The multi-word technique resembles the example based methods suggested recently. The technique primarily used thus far in SWETRA is a syntactical method based on Referent Grammar (cf. Sigurd 1987). The following extract from the lexicon shows multiple lexical entries (nominal, prepositional, and adverbial phrases) and single-word entries:

```
dlex(adv,m(finally,_),_,_,_,_,_,_) --> [til,sidst].
dlex(adv,m(to_use_when_mounting,_),_,_,_,_,_,_) -->
    [til,brug,for,montering].
dlex(p,m(to,_),_,_,_,_,_,_) --> [til].
dlex(adv,m(acc_to_pattern,_),_,_,_,_,_,_) -->
    [med,mønster,efter,diagrammet].
dlex(p,m(with,_),_,_,_,_,_,_) --> [med].
```

The lexicon of KNITTRA is sorted in reverse alphabetical order, which means that multiple lexical entries are stored before single words. When looking up lexical items in the text, larger chunks are found before single words. For example, the pattern *til sidst* is found before the single word *til*.

This means that two words are identified directly instead of only one at a time. This is done in order to achieve even better and faster results.

Symbols in the lexicon

As mentioned previously there are also some symbols used in knitting instructions. The symbols are included in the lexicon in the following way:

```
dlex(v,m(equal,pres),vt,fin,_,_,_,i,[]) --> ['='].
```

Done like this, the equals sign can be used as a predicate expression and hence it can be treated as any regular word. The following printout shows how a sentence including = is translated.

```
17 m og 21 p på p 5 i glst = 10 x 10 cm.
[subj(and1(s(m(indef,_6218),m(s(n_stitches,17),pl),[]),s(m(indef,_66D8),
m(s(n_rounds,21),pl),empty(s(m(on,_6BB0),s(m(indef,_6F00),m(s(needle_num
ber,5),sg),empty(m(in_flat_knitting,_7540),[]))),[])))),pred(m(m(equal,p
res),[])),obj(s(indef,s(10,x,10),m(centimeter_abbr,pl),[])),obj([]),advl
([]),advl([]),advl([]),advl([]),co(_53C8,[.])]
17 maskor och 21 varv på sticka nr 5 i slätst = 10 x 10 cm.
Used time: 0 m 7,3 s.
```

Multi-word entries with variables

Measurements frequently occur in knitting instructions and are interpreted as noun phrases. The numbers vary and in order to treat them effectively, the measurements are introduced in the lexicon with a variable right-hand side as shown below.

```
dlex(n,m(s(n3_stitches,[N1,N2,N3]),pl),indef,agr(indef,_,pl),_,_,_,_,i) -->
    [N1], {number(N1)}, ['('],
    [N2], {number(N2)}, [')'],
    [N3], {number(N3)}, [m].
```

Now consider the sentence *Strik 4 p glst på p nr 5*. For an experienced knitter, it is obvious that this means *Sticka 4 varv på sticka nr 5* ('Knit 4 turns on needle no 5'). In Danish, the *p* is ambiguous and may mean two things, both 'turn' and 'needle'. In the grammar section I will show how the problem can be solved syntactically. Here I will show how it can be solved using multi-word lexical items. The entries are:

```
dlex(n,m(s(n_turns,N),pl),indef,agr(pl,_,indef),_,_,_,_,i) -->
    [N], {number(N)}, [p].
dlex(n,m(s(needle_number,N),sg),indef,agr(sg,_,indef),_,_,_,_,i) -->
    ([p];[pind]), ([nr];['nr.'];[nummer]), [N], {number(N)}.
```

When these lexical items are used, the context helps to identify the different meanings of *p* as illustrated by the following printout.

```
Strik 4 p på p nr 5.
[pred(m(knit,imp)),obj(s(m(indef,_26B0),m(s(n_turns,4),pl),[])),obj([]),
advl([]),advl([]),advl(empty(s(m(on,_2AB0),s(m(indef,_2D58),m(s(needle_n
umber,5),sg),[])),[])),co([],[.])]
Sticka 4 varv på sticka nr 5.
Used time: 0 m 1,4 s.
```

The relevant parts of the representation are found in the underlined parts of the functional representation.

The morphology in KNITTRA

The morphology rules enable an extension of the lexicon. Not everything can be stored in a lexicon. Instead morphological rules are used to derive new forms when needed.

The morphological rules are the second level in the three-level hierarchy of KNITTRA (refer to the diagram in the introduction). The lexicon is almost never accessed directly from the grammar rules, but through the morphological rules using the predicate *dlexg*. If there were no morphological rules, the lexicon would have to contain all available forms of the words used – a technique which might be quick and convenient for small lexicons, but requires a lot of space. Instead of storing all forms, regular patterns in the language can be treated with morphological rules. The following is the default rule:

```
dlexg(A,B,C,D,E,F,G,H,I) --> dlex(A,B,C,D,E,F,G,H,I).
```

This means that an attempt is made to find the item searched directly in the lexicon. If it fails, the next step is to apply a morphological rule to the given word. The term *dlex* indicates the standard Danish lexicon, whereas the term *dlexg* refers to the generalised morphologically expanded Danish lexicon.

Suppose the word *masker* is searched for (Danish for *maskor* 'stitches'). This is what is available in the lexicon:

```
dlex(n,m(stitch,sg),indef,agr(sg,r,indef),_,_,e,r,r2) --> [maske].
```

To avoid storing information about the plural form *masker*, there is information in the entry for *maske* about the declension that should be used. In this particular case, the rule for declension is no. 2 (*r2* in the last argument).

The word *masker* is not stored in the lexicon, so the default morphological rule shown above will fail. The following morphological rule will solve the problem:


```
dlexg(n,m(B,pl),D,agr(pl,_,indef),_,_,_,R1,r2,[W1|Rest],Rest) :- % 1
   var(W1), dlex(n,m(B,sg),D,agr(sg,_,indef),_,_,_,R1,R,[W],_),  % 2
   (R=r2;R=r4), ccat(W,r,W1);                                    % 3
   nonvar(W1), ccat(W,r,W1),                                     % 4
   dlex(n,m(B,sg),D,agr(sg,_,indef),_,_,_,R1,r2,[W],_).          % 5
```

In line 1, the arguments tell us that we are looking for something in plural, indefinite form, and applying declension r2. What has to be done? Find the singular!

In lines 2 and 3 this is done at synthesis, and in lines 4 and 5 it is done when analysing a word. The final *r* is removed and the word *maske* is looked up in the lexicon. The word *maske* is found, and hence *masker* is the correct indefinite plural form of the indefinite singular *maske*. The predicate *ccat* is a string concatenation predicate.

The grammar in KNITTRA

The third and last part of KNITTRA, and other SWETRA applications, is the grammar. The grammar rules are explained and described in this section in parallel with the solution for the verb particles. An example of a sentence in the imperative with a verb particle is the following:

```
Slå 12 masker op.
[pred(m(cast_on,imp)),obj(s(m(indef,_42E0),m(s(n_stitches,12),pl),[])),o
bj([]),advl([]),advl([]),advl([]),co(_37B8,[.])]
Lägg upp 12 maskor.
Used time: 0 m 0,49 s.
```

This sentence is analysed using the following basic rule for sentences in the imperative:

```
ds(i,[],[pred(m(V,imp)),obj(N1),obj(N2),                          % 1
      advl(A1),advl(A2),advl(A3),co(V1,Fu)]) -->                   % 2
   {inom(Mi), itid(Mi)},                                           % 3
   dlexg(v,m(V,imp),Type,fin,Agr,_,_,_,PT), {Type\=aux, Aux=[]},  % 4
   {[P|E] = PT;P = PT, E = []},                                    % 5
   check_particle(P),                                              % 6
   dadvp(A1),                                                      % 7
   ({E = [deemph]}, dopc(P);{E = [emph];P = []}),                  % 8
   ({Type = dvt}, dcnpo(Agr,N1), dcnpo(Agr,N2);                    % 9
    {Type = vt}, dcnpo(Agr,N1), {N2 = []};                         %10
    {Type = vi, N1 = [], N2 = []}),                                %11
   dadvp(A2),                                                      %12
   ({E = [emph]}, dopc(P);{E = [deemph];P = []}),                  %13
   dadvp(A3),                                                      %14
   dscoord(i,V1,Fu,Type).                                          %15
```

Lines 1 and 2 describe the mode, topic, and functional representation of a sentence. The first argument *i* indicates that this is the structure of a sentence in the imperative. The second argument *[]* is used for the topic in SWETRA rules, but is not used in KNITTRA. The third argument is a list containing the functional roles in the analysed sentence. In this case there is a predicate *(pred)*, place for two objects *(obj)*, three adverbials *(advl)* and a conjunction *(co)* with a possible second clause whose functional representation would be found in the variable *Fu*.

The predicates in line 3 are used for timing. The first one, *inom(Mi) (within(Mi))*, retrieves a time pointer from the database, and the second, *itid(Mi) (in_time(Mi))*, checks whether the time allowed is consumed or not. These two predicates do not affect the analysis directly. They only prevent the system from processing a sentence for too long. If the allowed time is consumed, the predicate *itid* will fail and, hence, the syntactic analysis will stop and one of the two other methods will be applied, the phrasal and eventually the word-for-word approach.

Since this rule is for a sentence in the imperative, the first thing to do is to find a verb in the imperative as the first element in the sentence. This is done in line 4. Note that the search is done using the morphological rules as explained in the previous section. This predicate returns information about the transitivity of the verb (*Type*) and what type of particle (*PT*) is attached to the verb, if any. The particle can be of two types, which is reflected in line 5. The first case says that there is a particle attached to the verb, and that it is either stressed or unstressed (*E*). In the second case there is no particle, so the stress is set to nil (or the empty list *[]*). The next step is to check whether the particle is present in the rest of the sentence. If it is not found, the predicate *check_particle* will fail, and a new lexical item must be looked for. This predicate speeds up the parsing.

The next part of the sentence might be an adverbial. If it is missing, the corresponding variable is set to nothing, i.e. *A1* has the value *[]* after the call to *dadvp(A1)*. In line 8 the verb particle is checked for. If an unstressed particle was found in line 5, it should be in this position.

In lines 9-11, different number of objects are expected depending on the verb type found in line 4.

In line 12 there is another call for an adverbial, (*A2*). The same rules apply here as for the call to *dadvp(A1)*.

In line 13 there is a check for whether the particle was stressed or not. If it was found stressed in line 5, the particle should then be found here.

In line 14 there is another place for an adverbial, (*A3*) and, finally, in line 15 there is a check for coordination.

In the example above, the Danish stressed particle *op* is placed after the object, whereas the corresponding particle in Swedish, *upp*, is placed before

the object in the Swedish grammar. Information about particles is stored in the lexicon in SWETRA. In order to find verbs with a particle before those without, the lexicon is sorted in reverse alphabetical order. This was described in an earlier section of this paper.

Definiteness

Another difference between Swedish and Danish mentioned above is shown in the title of the fairy tale *Den grimme ælling* by H.C. Andersen. In Swedish, this is called *Den fula ankungen* 'The ugly duckling'. Note that the Swedish title contains two modifiers (*Den* and *-en*), whereas the Danish title has only one (*Den*). The rule is that if a definite article is found, the succeeding noun must be in its basic form. The Danish grammar rule for this is:

```
% Example: den lille mand (the little man)
dnp(agr(Num,G,def),s(D,A,S,T)) -->
   dart(agr(Num,G,def),D),
   dap(agr(Num,G,def),A),
   dlexg(n,S,C,agr(Num,G,indef),_,_,_,_,_), {C\=pron},
   dpattr(S,T).
```

and the Swedish counterpart is:

```
% Example: den lilla mannen (the little man)
np(agr(Num,G,Def),s(D,A,S,T)) -->
   art(agr(Num,G,Def),D),
   ap(agr(Num,G,Def),A),
   slexg(n,S,C,agr(Num,G,Def),_,_,_,_,_), {C\=pron},
   pattr(S,T).
```

The difference is found in the agreement condition in the fourth line of each predicate. The Swedish equivalent requires the noun to be definite in combination with the definite article. The following translation is an example:

```
Strik den anden side færdig på samme måde , blot modsat.
[pred(m(knit,imp)),obj(s(m(def,sg),m(other,_4B04),m(side,sg),s(m(ready,_
4C1C),[],[]))),obj([]),advl([]),advl([]),advl(comma(m(in_the_same_way,_5
00C),empty(m(but_reversed,_53EC),[]))),co([],[.])]
Sticka den andra sidan färdig på samma sätt , fast motsatt.
Used time: 1 m 15,8 s.
```

Passive constructions

In Danish, the verb *være* 'be' is used in perfect passive constructions, while *hava* 'have' is used in Swedish as mentioned above. Another difference is that the participle with *blive* is not inflected in Danish, the neuter form is always used. In Swedish, the participle must be inflected depending on the subject's number and gender. Hence, passive constructions must be treated specially. Swedish may use *bliva* or *s*-passive as in: *12 m har blivit stickade* and *12 m har stickats* Danish has no passive participle, but an alternative expression: *12 m er blevet strikket* and *12 m er strikket.* In the second Danish sentence, the auxiliary verb *blevet* is missing, but the meaning is still the same. The Danish lexicon contains the two auxiliaries used for passive:

```
dlex(v,m(perf,pres),aux,fin,_,pass,_,i,[]) --> [er].
dlex(v,m(passive,nonf),aux,part,_,part,_,i,[]) --> [blevet].
```

The Swedish equivalent is then:

```
slex(v,m(perf,pres),aux,fin,_,pass,_,i,[]) --> [har].
slex(v,m(passive,nonf),aux,part,_,part,_,i,[]) --> [blivit].
```

In the morphological rules, the following rule is present:

```
dlexg(v,[m(B,nonf),m(passive,pres)],vi,fin,C,D,E,_,P,[W1|Rest],Rest) :-
   var(W1), dlexg(v,m(B,nonf),vt,inf,C,D,E,_,P,[W],_), ccat(W,s,W1);
   nonvar(W1), ccat(W,s,W1),
   dlexg(v,m(B,nonf),vt,inf,C,D,E,_,P,[W],_).
```

This rule says that whenever a verb ending with *s* is found and where the form without *s* is an infinitive, it should be treated as a verb in passive form. It also says that it is only applicable for transitive verbs and that the resulting verb form is intransitive.

In Danish, it is not necessary for the verb *blevet* to be present, as described in one of the two examples above. The examples are solved using these grammar rules:

```
dclex(v,[m(V,nonf),m(passive,nonf)],vi,pass,Agr,_,C,D,PT) -->
   dlexg(v,m(passive,nonf),aux,part,_,Dem,_,_,_),
   dlexg(v,m(V,nonf),vt,Dem,_,_,_,_,PT).
dclex(v,[m(V,nonf),m(passive,nonf)],vi,pass,Agr,_,C,D,PT) -->
   dlexg(v,m(V,nonf),vt,part,_,_,_,_,PT).
```

The first structure treats the case with *blevet*, and the second without. The first rule is also present in the Swedish grammar as the Swedish equivalent can be solved in the same way, but the second is not.

Solving ambiguity syntactically

Consider the sentence *Strik 4 p glst på p nr 5.* again. If the multi-word lexical items are not introduced in the system, then this has to be solved syntactically. In the instructions, the word *p* is only found in the positions shown above. Therefore, the treatment of the ambiguity problem is solved in two steps, one lexical and one syntactic step. The lexical step involves

marking the word for 'turns' as a unit, as it is always used in the plural. The word for 'knitting needle' is then marked as a normal indefinite noun as shown here:

```
dlex(n,m(turn_abbr,pl),unit,agr(pl,r,_),_,_,_,_,i) --> [p].
dlex(n,m(knitting_needle_abbr,sg),indef,agr(sg,r,indef),_,_,_,_,i) --> [p].
```

This is not enough though. The syntactic rules for solving this are shown below. The unit meaning is used as an adjective-like expression (indicated by the predicate name *da*, Danish adjective):

```
da(Agr,s(N,U)) --> [N], {number(N), N > 1},
                   dlexg(n,U,unit,agr(pl,_,indef),_,_,_,_,_),
                   {Agr = agr(sg,_,indef)}.
```

The second *p* in the example is then treated as a noun, and the following *no 5* is interpreted as an attribute to the second *p*. This is handled by the rule for post attributes.

```
dpattr(S,s(number,N)) -->
   dlexg(n,m(number,_),indef,agr(sg,_,indef),_,_,_,_,_),
   [N], {number(N)}.
```

Using these rules, the translation is the same as in a previous section, but the sub-structures (underlined) are different.

```
Strik 4 p på p nr 5.
[pred(m(knit,imp)),obj(s(4,m(turn_abbr,pl))),obj([]),advl([]),advl([]),a
dvl(empty(s(m(on, 2874),s(m(indef, 2B1C),m(knitting_needle_abbr,sg),s(nu
mber,5))),[])),co([],[.])]
Sticka 4 v på sticka nr 5.
Used time: 0 m 28,61 s.
```

Sentence termination

After a given instruction, a full sentence in the imperative often ends with an equal sign and some figures denoting the number of stitches or centimetres required in order to keep the size. These endings are introduced directly in the grammar as shown below. A sentence, for which this technique is used, is found in the last section of this paper, where an automatically translated text is shown.

```
scoord(M,T,['.'],_) --> [].                 % 1 End of sentence
scoord(M,T,['.'],_) --> ['.'].              % 2 End of sentence
scoord(M,T,[':'],_) --> [':'].              % 3 End of sentence
scoord(M,T,['!'],_) --> ['!'].              % 4 End of sentence
scoord(M,T,m(which_is,N),_) -->             % 5 End of sentence
   ['='], np(_,N), ['.'].                   % 6 with "equal phrase"
scoord(M,T,Coord,_) -->                     % 7 Coordinated sentence
   co(C),                                   % 8 Coordinator found
   {C \= empty, functor(Coord,C,1),
```

```
   arg(1,Coord,F)),               %  9 Build a structure
   check_s_finite,                % 10 Look for finite verb
   ss(M,T,F).                     % 11 Full sentence
```

Phrasal translation

The main idea of doing a phrasal translation is to try to translate larger chunks of text like nominal phrases, adverbial phrases, predicate constructions, and coordinators. Using this approach, it is possible to utilise gender and number when identifying nominal phrases. The method applied in KNITTRA is described by the following code extract. The first predicate below (*struct_ww*) contains the main predicate for the phrasal translating method, where the first argument is the number of seconds allowed for this type of translation before the word-for-word method is applied. The second argument is the Danish source sentence and the third argument is the Swedish target sentence. The second predicate (*struct_ds*) contains the translating predicate, from which calls are made to the specific structures looked for.

```
struct_ww(Sec,DS,SS) :-                                 %  1 Main predicate
   constrain(Sec),                                      %  2 Max time
   start(T1,M1,S1,H1),                                  %  3 Start timer
   struct_ds(DS,SS),                                    %  4 Translate
   stop(T2,M2,S2,H2),                                   %  5 Stop timer
   timing(T1,M1,S1,H1,T2,M2,S2,H2).                     %  6 Calculate time
struct_ds([],[]).                                       %  7 End of sentence
struct_ds(['.'],['.']).                                 %  8 End of sentence
struct_ds([':'],[':']).                                 %  9 End of sentence
struct_ds(['!'],['!']).                                 % 10 End of sentence
struct_ds(['?'],['?']).                                 % 11 End of sentence
struct_ds(DS,SS) :-                                     % 12 Translate phrases
   (dnp(_,N,DS,DR), np(_,N,SS,SR);                      % 13 Nominal phrase?
   dadv(A,DS,DR), adv(A,SS,SR);                         % 14 Adverbial phrase?
   dco(C,DS,DR), C \= empty, co(C,SS,SR);               % 15 Coordinator?
   dlexg(v,M,_,_,_,_,_,_,[DP,_],DS,DR1),                % 16 Verb with
   append(D1,[DP|RD],DR1), append(D1,RD,DR),            % 17 particle?
   slexg(v,M,_,_,_,_,_,_,SP,SS,SR1),                    % 18 Move the particle
   append(S1,[SP|RS],SR1), append(S1,RS,SR);            % 19 forward
   dlexg(v,M,_,_,_,_,_,_,[],DS,DR),                     % 20 Verb without
   slexg(v,M,_,_,_,_,_,_,[],SS,SR);                     % 21 particle?
   append([E],DR,DS), append([E],SR,SS);                % 22 Unknown item
   fail), !,                                            % 23 This didn't work.
   struct_ds(DR,SR).                                    % 24 Continue
```

Lines 1-5 are briefly commented on in the code and further clarified here. Lines 2, 3, and 5 are for timing purposes only and are not a part of the linguistic treatment. Lines 1 and 4 are the 'linguistic' parts where the sentences and structures are defined and used. Lines 7 to 11 contain the different sentence endings that are available. Unless the sentence endings are found, the process continues in line 12 where the first argument describes

the Danish sentence and the second the translated Swedish sentence. The following lines are attempts at solving larger parts of the source sentence.

In line 13, an attempt is made to find a noun phrase. If found, it is immediately translated into Swedish. If a noun phrase is not found, the next attempt, in line 14, is to find an adverbial phrase. Line 15 looks for a coordinator (unless it is not the empty coordinator). Lines 16-19 take care of the particle verbs. In this case, the particle is removed from where it is located in the sentence and moved to the position immediately following the verb. In case there is no particle, a verb without a particle is assumed (lines 20 and 21). In the case nothing applicable is found, line 22 is used to transfer the unknown part to the target sentence and then continue with the rest.

Handling new words

The most interesting step in the work with KNITTRA was feeding the system with new empirical material after a period of implementing the lexicon, the morphology, and the grammar rules. Most of the new sentences were solvable syntactically, but many words were not present in the lexicon. These words were normally proper names, e.g. yarn brands, and colours. The colours had to be added to the lexicon as opposed to the yarn brands, which can be handled with the pre-processor. These topics are discussed below.

Colours

When adding the colours to the lexicon, they have to be added both as an adjective and a mass noun. The colours are used as references to the yarn. One can say that the word for 'yarn' is missing, and the colour name takes that role as well, which is the reason for storing them as mass nouns as well as adjectives. A typical use of the colour as a mass noun is shown in the following example:

Slå 180 - 189 - 195 m op på rundp 3.5 med rødt, og strik 5 cm ribkant, skift efter 2 p til koral.
'Cast 180 - 189 - 195 stitches on on circular-needle 3.5 with red, and stitch 5 cm row-edge, change after 2 turns to coral.'

And the lexical entries needed are:

```
dlex(n,m(red_yarn,mass),_,agr(_,n,_),_,_,_,_,i) --> ['rødt'].
dlex(n,m(coral_yarn,mass),_,agr(_,n,_),_,_,_,_,i) --> [koral].
dlex(a,m(red,_),_,agr(sg,_,indef),_,_,_,_,_) --> ['rød'].
dlex(a,m(coral,_),_,agr(_,_,_),_,_,_,_,_) --> [koral].
```

Translation example:

```
Slå 180 - 189 - 195 m op på rundp 3.5 med rødt , og strik 5 cm ribkant ,
skift efter 2 p til koral.
[pred(m(cast_on,imp)),obj(s(m(indef,_0C7C),m(s(n3_stitches,[180,189,195]
),pl),[])),obj([]),advl([]),advl([]),advl(empty(s(m(on,_0E08),s(m(indef,
_0E38),m(s(round_needle_number,3.5),sg),empty(s(m(with,_1170),s(m(indef,
_11A0),m(red_yarn,mass),[])),[]))),[])),co([],and3([pred(m(knit,imp)),ob
j(s(indef,s(5,m(centimeter_abbr,pl)),m(row_edge,sg),[])),obj([]),advl([]
),advl([]),advl([]),co([],comma([pred(m(switch,imp)),obj([]),obj([]),adv
l(empty(s(m(after,_20CC),s(m(indef,_20FC),m(s(n_rounds,2),pl),empty(s(m(
to,_22A0),s(m(indef,_22D0),m(coral_yarn,mass),[])),[]))),[])),advl([]),a
dvl([]),co(_1BA0,[.])])))])]
Used time: 0 m 10,38 s.
Lägg upp 180 ( 189 - 195 ) maskor på rundsticka nr 3.5 med rött , och
sticka 5 cm ribbkant , byt efter 2 varv till korall.
```

Note the mass nouns in the functional representation.

Proper names

New proper names do not have to be added to the lexicon, as they are normally not translated anyway. Instead, they can be replaced by placeholders in the sentence during the translation pass. The problem with proper names is not limited to knitting instructions. When translating stock market reports for example, the company names are not translated.

KNITTRA will pre-process every sentence before the translation pass is started. Every word is checked for an initial upper-case character. If such a character is found, it is replaced with the equivalent lower-case and then the lexicon is searched. If a match is found, this word will be left as is in the sentence (in lower-case though). If, however, the word is not found, it is considered a proper name and is replaced with a placeholder (*gnrl_prop*) in the sentence to translate. After the translation pass, the placeholders are replaced with the original proper names. This means that an initial word in a sentence only will be replaced with a placeholder for a proper noun if the word is not found in the lexicon.

The reason for only checking for upper-case characters is that the lexical entries in KNITTRA are all stored in lower-case characters.

The same technique is used for embedded comments, e.g. phrases within parenthesis. The embedded comments are treated as adverbials and the placeholder is in this case called *gnrl_advl*.

An example of both these features is shown below.

```
Der er anvendt ulakerede træperler samt rester af olivengrøn mintgrøn ,
lys blå og lys rød Muskat Soft ( 100 % bomuld , løbelængde ca. 100 m /
50 g ) fra Garnstudio eller tilsvarende kvalitet.
[subj(and2(s(indef,m(unvarnished,_132C),m(wooden_button,pl),[]),s(m(inde
f,_14FC),m(remnant,pl),empty(s(m(of,_1738),s(indef,empty(m(olivegreen,_1
```



```
908),comma(m(mintgreen,_19D0),empty(m(light,_1A98),m(blue,posit) and
s(m(light,_1C38),m(red,posit))))),m(gnrl_prop,_1CB0),m(gnrl_prop,prop)))
,empty(m(gnrl_advl,_20D8),empty(s(m(from,_24C4),or1(s(m(gnrl_prop,_2A84)
,[]),s(indef,m(corresponding,_2F74),m(quality,sg),[]))),[])))))),pred(m(
[m(use,nonf),m(passive,nonf)],m(perf,pres))),obj([]),obj([]),advl([]),ad
vl([]),advl([]),advl([]),co(_0AB8,[.])]
Used time: 0 m 16,37 s.

Olackerade träpärlor samt rester av olivgrön mintgrön , ljus blå och
ljus röd Muskat Soft ( 100 % bomull , garnvikt ca 50 g / 100 m ) från
Garnstudio eller motsvarande kvalitet har använts.
Used time: 0 m 20,87 s.
```

Note the two types of placeholders in the functional representation. The first time printed is the time used for the syntactic analysis and the second time shows the total time used for the translation, including replacing the placeholders with the translated adverbials or the proper names.

References

Karlgren, H. 1988. 'Term-tuning, a method for the computer-aided revision of multi-lingual texts'. In *International Forum on Information and Documentation* 13:2 (April 1988), 7-13.

Kristensen, K. & L. Brink. 1986. *Dansk for svenskere*. Malmö: Liber.

Nielsen, N. & P. Lindegård Hjorth. 1959. *Dansk sproglære for svenskere*. Lund: Gleerups.

Sigurd, B. 1987. 'Referent Grammar (RG). A generalized phrase structure grammar with built-in referents'. *Studia Linguistica* 41:2, 115-35.

Sigurd, B. 1993. 'The problems of multi-word prepositions and subjunctions'. *Working Papers* 40, 197-212. Department of Linguistics, Lund University.

Sigurd, B., M. Eeg-Olofsson, B. Gawrońska-Werngren & P. Warter. 1990. *SWETRA – a multilanguage translation system* (=*Praktisk Lingvistik* 14). Department of Linguistics. Lund University.

Sigurd, B., M. Eeg-Olofsson, C. Willners & C. Johansson. 1992. *Automatic translation in specific domains: weather (Weathra) and stock market (Stocktra, Vectra)* (=*Praktisk Lingvistik* 15). Department of Linguistics, Lund University.

Appendix

Example of a text translated automatically.

The text below took approximately 22 minutes to translate, which corresponds to less than one minute per sentence.

Danish Source Text	*Translated Swedish Text*
Strikkefasthed:	Masktäthet:
17 m og 21 p på p 5 i glatstrik = 10 x 10 cm.	17 maskor och 21 varv på sticka nr 5 i slätst = 10 x 10 cm.
Ribkant:	Ribbkant:
* 1 r, 2 vr *.	* 1 r, 2 avig *.
Ryg og forstykke:	Rygg och framstycke:
Jakken strikkes frem og tilbage på rundp og deles ved ærmegabet.	Kavajen stickas fram och tillbaka på rundsticka och delas vid ärmhålet.
Slå 180 - 189 - 195 m op på rundp 3.5 med rødt, og strik 5 cm ribkant, skift efter 2 p til koral.	Lägg upp 180 (189 - 195) maskor på rundsticka nr 3.5 med rött, och sticka 5 cm ribbkant, byt efter 2 varv till korall.
Skift til rundp 5, samtidig med at der tages 24 - 27 - 33 m ud jævnt fordelt på 1. p = 204 - 216 - 228 m.	Byt till rundsticka nr 5 samtidigt som 24 (27 - 33) maskor tas ut jämnt fördelat på 1:a stickan = 204 (216 - 228) maskor.
Husk strikkefastheden!	Kom ihåg masktätheten!
Strik mønster således (mønster 1 skal måle 18 cm i højden, hvis ikke, så afpas efter det glatstrikkede):	Sticka mönster på detta sätt (mönster 1 skall mäta 18 cm i höjden, om inte, så avpassa efter det slätstickade):
5 - 6 - 1 cm glat , mønster 1 , 7 - 9 - 9 cm glat.	5 (6 - 1) cm slätst, mönster 1 , 7 (9 - 9) cm slätst.
Str small og medium:	Stl small och medium:
Mønster 2, mønster 3, strik færdig med glat.	Mönster 2, mönster 3, sticka färdigt med slätst.
Str large:	Stl large:
Strik færdig med mønster 1.	Sticka färdigt med mönster 1.
Samtidig strikkes:	Samtidigt stickas:
Når arb måler 40 - 45 - 50 cm , deles arb således:	När arb mäter 40 (45 - 50) cm delas arb på detta sätt:
50 - 53 - 56 m forstykke, luk 2 m af, 100 - 106 - 112 m ryg, luk 2 m af, 50 - 53 - 56 m forstykke.	50 (53 - 56) maskor framstycke, maska av 2 maskor, 100 (106 - 112) maskor rygg, maska av 2 maskor, 50 (53 - 56) maskor framstycke.
Hver del strikkes færdig for sig.	Vart stycke stickas färdigt för sig.
Ryg:	Rygg:
= 100 - 106 - 112 m.	= 100 (106 - 112) maskor.
Fortsæt mønsteret.	Fortsätt mönstret.
Luk til ærmegab på hver 2. p for 2 m x 2, 1 m x 2 - 1 - 2 = 88 - 96 - 100 m.	Maska till ärmhål på varannan sticka av 2 maskor x 2 , 1 maska x 2 (1 - 2) = 88 (96 - 100) maskor.

Lund University, Dept. of Linguistics
Working Papers 42 (1994), 159-178

The Data Behind the Elsewhere Condition

Magnus Olsson

Kiparsky 1973 proposes the Elsewhere Condition, which is intended to pick out which pairs of rules are disjunctively ordered. He offers certain cases where his condition seemingly supersedes the *SPE* approach. In this paper, I re-examine the data from Finnish, Karok, Diola-Fogny and Sanskrit (both high-segment syllabicity and external sandhi). It appears that the Elsewhere Condition is not a necessary device for obtaining the right output and, also, that the real data lying behind the examples in each case constitutes no problem for the *SPE* approach. It is concluded that the theory of disjunctivity outlined by Kiparsky can not be shown to be superior to the *SPE* approach – at least, given the examples.

Foreword

It is a truism that conditions in linguistics are more abstract than the data. This ought to mean that the data are more basic, although there is not universal agreement on this point. In fact, in a casual conversation with another linguist while attending a summer school, I was amazed at her reaction to my work on checking Kiparsky's 1973 examples in favour of the Elsewhere Condition. She did not care about the relevance of the data – although gathered in support of a theoretical proposal – stating that these were just examples to show the workings of the condition. One may wonder whether this is a widespread view. The want of critical studies of proposed theoretical devices, centering on the real nature of the supporting evidence, adds weight to my thoughts in that direction. The present case does not appear to be an isolated instance either. Much energy has been devoted to the production of new general hypotheses while the true facts about the data at hand have sometimes dwelt elsewhere.

The examination of a case where an author appealed to the Elsewhere Condition first directed my attention to the proposal. Soon the actual nature of the arguments that Kiparsky adduces in favour of his condition aroused my interest. The question was whether, in light of the data, the conclusion was inevitable that the condition was superior to the *SPE* approach or whether there were other solutions. Once in focus, the project of solving the riddles – as I saw them – developed into an irresistible temptation. The