Lund University, Dept. of Linguistics
Working Papers 51 (2005), 175–186

Segram – a program visualizing stages of grammatical competence

Bengt Sigurd, Gisela Håkansson and Johan Dahl

Introduction

Segram is a computer program constructed to be used in the teaching of courses in grammar and language learning. It is available at the Department of Linguistics, Lund University and can be demonstrated by the authors. It is originally written in Prolog (LPA-Prolog) for Macintosh. It supplements the educational program *Gramte* (Lastow & Håkansson 1997).

Segram demonstrates the grammatical analysis of Swedish sentences as tree diagrams, parenthesis representations and English translations. Following traditional Swedish grammar, *Segram* shows two syntactic diagrams, one based on word/phrase classes/categories (Swedish: *ordklasser)* and one based on functional roles *(satsdelar).* The trees correspond to parenthesis representations, which are also presented. The functional diagram furthermore shows the meanings of the words given in a kind of Machinese English and the functional representation can therefore be used as an interlingua in automatic translation (see Sigurd 1994).

Segram visualizes mainly grammatical structures. The vocabulary is intentionally very restricted, but the program contains a wide range of syntactic rules. Naturally *Segram* can only analyze sentences containing words included in its lexicon and only sentences using grammatical constructions included in its grammar.

Segram is constructed to reflect stages in language learning according to Processability theory (Pienemann 1998, Pienemann & Håkansson 1999) and it simulates learner language at different stages by grammars with different sets of rules. The command *analyze1(X),* where *X* is a string of words, can only analyze sentences which according to Processability theory are available at the first level. The command *analyze2(X)* simulates the second level of processing, etc. Seven levels are assumed. *Segram* can also generate sentences. The

command *generate3* can generate sentences characteristic of the third level (which includes the first and second levels).

As a first demonstration one may write the commands: *demo1, demo2, ... demo30* to the program and watch the resulting analyses.

The lexicon

The present lexicon of *Segram* is restricted for pedagogical reasons, but it can, of course, be extended. The program contains one intransitive verb *sprang* 'ran', one transitive verb *bet* 'bit', one belief verb *trodde* 'thought', one predicative verb (copula) *var* 'was' and one auxiliary verb *kunde* 'could'. All finite verbs in *Segram* are in the past tense, but there are also two infinite verb forms *bita* 'bite' and *springa* 'run', to be used with the auxiliary *kunde*.

There are several types of adverbials in *Segram*: a sentence (belief) adverbial (Swedish: *satsadverbial*) *troligen* 'probably', one manner adverbial *(sättsadverbial) snabbt* 'quickly', which can be modified by the adverb adverb *mycket* 'very' as in *mycket snabbt* 'very quickly', one time adverbial *(tidsadverbial) igår* 'yesterday', one place adverbial *(platsadverbial) på gatan* 'in the street', one negative adverbial *inte* 'not' and the meta comment adverbial *minst sagt* 'to say the least'. A subordinate clause introduced by *när* 'when' can also serve as a time adverbial. *Segram* only accepts two adverbials per sentence, in subordinate sentences generally no adverbials.

There is only one adjective *snäll* 'nice' which may be an attribute in the form *snälla* in definite noun phrases: *den snälla hunden* 'the nice dog', *det snälla barnet* 'the nice child'. Used predicatively, an adjective has to be inflected according to gender and number in Swedish: *barnet var snällt, hunden var snäll.* An adjective can also be determined by the adverb (adjective adverb) *mycket* 'very' in *Segram.*

For pedagogical reasons the set of nouns is restricted to *barnet* 'the child', *hunden* 'the dog', the question pronoun *vem* 'who' and the personal pronoun *hon* 'she'/*henne* 'her'. *Segram* includes no plural forms.

A noun phrase can only have one attribute beside the article: *den snälla hunden* 'the nice dog'. The attribute may be a post attribute, either a prepositional phrase: *hunden på gatan* 'the dog in the street' or a relative clause, as in *barnet som hunden bet* 'the child that the dog bit'. A relative clause may be a subject relativized clause: *hunden som sprang* 'the dog that ran' or *hunden som bet barnet* 'the dog that bit the child'. The object relativized clause is illustrated by *barnet som hunden bet* 'the child that the dog bit'. Only some simple relative clauses without adverbials can be analyzed

in the present version of *Segram.* The program also includes the subjunction *att* 'that' introducing subordinate (object) clauses with *trodde* 'thought'.

Commands

Segram analyzes sentences in general by the command *analyze(X)*, where X is a list of words spelled with no initial capitals and separated by commas as in *analyze*([den, snälla, hunden, bet, inte, barnet, idag]). If the analysis is to be restricted to a certain level the commands are: *analyze1(X) ... analyze7(X).*

It is also possible to make *Segram* generate sentences. The command *generate1* will produce sentences acceptable by the first level grammar, *generate2* will generate sentences according to the second (and first) level, etc. The highest generate command is *generate7*, which produces all sentences acceptable at this high level, which includes all the sentences produced at the lower levels (except sentences without inversion after an initial adverbial occurring at the third level). The program can generally give several solutions. One gets the first solution from the program by pressing the button *first,* next by pressing the button *next.* If the button *all* is pressed the computer will generate all sentences of that level – which often will be numerous.

It is also possible to translate between Swedish and English by *Segram* using the command *setrans(X,Y).* Writing *setrans*([hon,sprang,inte,snabbt],*Y*) will result in an English translation ([she,did,not,run,quickly]) in the variable *Y.* There is a parallel English grammar in *Segram* with corresponding words and syntactic rules. The English grammar does not have to handle inversion of subject and predicate if an adverbial introduces the sentence, as the Swedish grammar does, but it has to handle the complications of *do*-support in negated sentences and questions. The following figures (1 and 2), representations and translation show the analysis of *Barnet trodde inte att hunden som sprang var snäll.*

Barnet trodde inte att hunden som sprang var snäll

decl
level5
[subj([the, m(child, sg)]), pred(m(think, past)), obj([subj([def(def), h(m(dog, sg)), postattr([subj([def(def), h(m(dog, sg))]), pred(m(run, past))])]), pred(m(be, past)), predikativ(nice)]), nadvl(not)]

[subjnomfras(n(barnet)), verbfras(v(trodde)), negadv(adv(inte)), obj(subjunk(att), bisats([subjnomfras([nomfras(n(hunden), postattr(rel(som), verbfras(sprang)))]), verbfras(v(var)), adjfras(a(snäll))])]

The child did not think that the dog that ran was nice

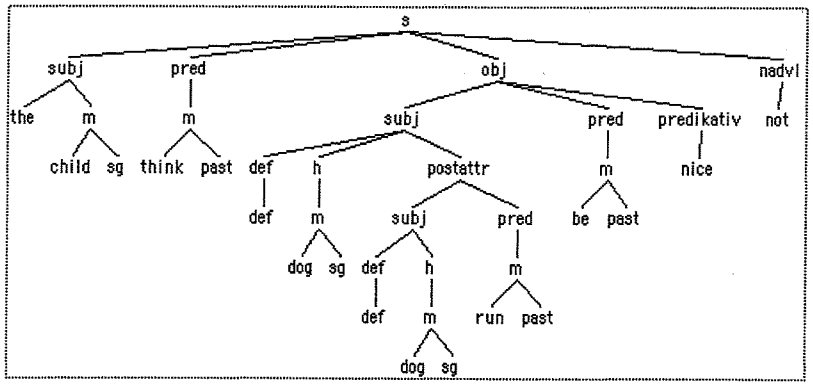

Figure 1. Functional diagram

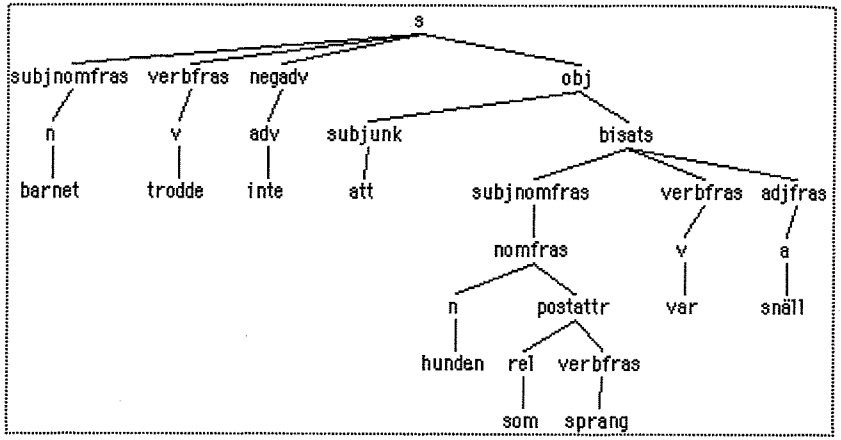

Figure 2. Phrase structure diagram

The rules at the different levels

As mentioned, rules are included or excluded simulating different levels of grammatical competence. The levels programmed are based on Pienemann 1998 and Pienemann & Håkansson 1999. The rules at the lower levels are generally included in the higher levels.

An optional rule is incorporated in the grammar program with a condition, e.g. the following which states that the sentence adverb *troligen* 'probably' is included only if *Segram* is ordered to include the fact: *lex(trol),* which is a

characteristic of the sixth level. A condition is written inside curly brackets in Prolog.

sadv(probably, sadv) --> {lex(trol)}, [troligen].

Similarly, if the condition {*lex*(som))} is fulfilled the system accepts relative clauses, as in *barnet som hunden bet*, *hunden som bet barnet* and *hunden som sprang*. Such relative clauses are not allowed until level 5.

Level 0 – no grammar
This level is characterized by the lack of linguistic structure, no morphology, no grammar, i.e. no word inflection, no phrases, no fixed word (phrase) order, no agreement, no subject-predicate structure.

Level morph
This pre-grammar level is characterized by some generalized, but not standard Swedish word inflections such as plural *-ar,* e.g. *lammar* 'lambs' (normally *lamm*), *husar* 'houses' (normally *hus*), plurals of *lamm* and *hus,* respectively. Similarly, verbs are often inflected for past tense by adding *-de (-dde* after stressed vowel) as in *fallde* 'fell', *springde* 'ran', *gådde* 'went' (to *gå*), *tadde* 'took' (to *ta*), *fådde* 'got' (to *få*). The existence of two different inflection patterns is evidence of the existence of two categories in the learners: verbs and nouns.

Level 1
This is the first grammatical level with emerging targetlike nominal and verbal inflection, but only simple noun phrases occur, as in *Hund sprang*.

Level 2
This level is characterized by noun phrases with agreement: *(Den) snälla hunden sprang*.

Level 3
This level allows copulative sentences with *var* with agreement *(Barnet var snällt, Hunden var snäll).* Transitive verbs *(bet)* with objects are allowed *(Hunden bet barnet).* Auxiliaries *(kunde)* with infinitives are also accepted. Some second language learners (but not first language learners) accept preposed (topicalized/focused) adverbials of time *(igår),* place *(på gatan)* and

manner *(snabbt)* with non-inverted word order at this level, e.g. *Igår hunden bet barnet.*

Level 4
This level allows preposed adverbs of time *(igår)*, etc., only with inverted word order *(Igår bet hunden barnet)*. Post-attributed prepositional phrases *(hunden på gatan)* are also accepted. The non-inverted word order acceptable with some learners at level 3 is not accepted at level 4. Inverted word order is used in yes/no questions at this level: *Bet hunden barnet?* 'Did the dog bite the child?'.

Level 5
Subordinated clauses are a characteristic of this level. Level 5 allows complex noun phrases with relative clauses *(pojken som hunden bet)*. The functional representation of a relative clause repeats the (underlying) correlate. Subordinated *att* clauses are allowed with the verb *trodde*. Subordinated clauses are accepted as adverbials, e.g. with the temporal subjunction *när* 'when' as in *när hunden sprang*.

Level 6
The sentence adverbial *troligen* 'probably' which comments on the truth of the sentence is acceptable and regarded as a high degree of competence.

Level 7
The metacommenting adverbial phrase *minst sagt* 'to say the least' is a characteristic of level 7 and taken as evidence of a very high degree of grammatical and cognitive competence.

The commands will not accept sentences above the competence indicated by the numbers. The command *analyze1*([hunden, bet, barnet]) will fail, but *analyze3*([hunden, bet, barnet]) will be successful and result in a functional representation and tree, a phrase structure representation and tree, a mode label (declarative, decl) and an English translation in the Output window. The sentence [sprang, den, snälla, hunden] will not be accepted below *analyze4* as it includes a question with inverse word order. The command *analyze5* can analyze [hunden, trodde, att, hunden, som, sprang, bet, barnet, som, barnet, bet, när, hunden, sprang, igår].

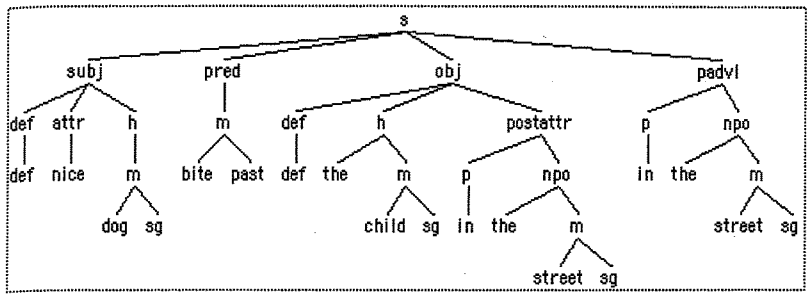

Figure 3. Functional diagram

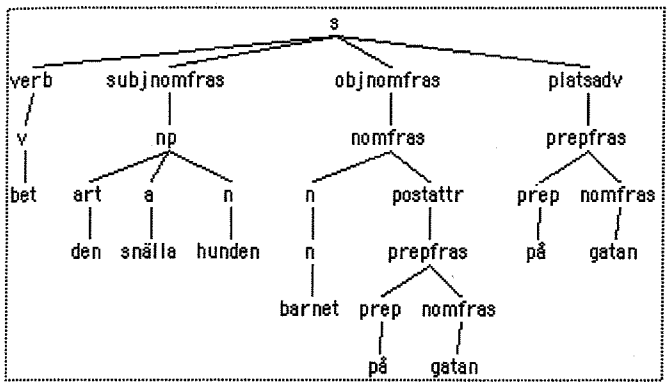

Figure 4. Phrase structure diagram

The meanings and pragmatics of the sentences may sometimes be strange as also shown by the example *Bet den snälla hunden barnet på gatan på gatan*. Note the different functions of the preposition phrase *på gatan* (see Figures 3 and 4).

Bet den snälla hunden barnet på gatan på gatan

quest
level4
[subj([def(def), attr(nice), h(m(dog, sg))]), pred(m(bite, past)), obj([def(def), h([the, m(child, sg)]), postattr([p(in), npo([the, m(street, sg)])])]), padvl([p(in), npo([the, m(street, sg)])])]

[verb(v(bet)), subjnomfras(np(art(den), a(snälla), n(hunden))), objnomfras([nomfras(n(n(barnet)), postattr(prepfras(prep(på), nomfras(gatan))))]), platsadv(prepfras(prep(på), nomfras(gatan)))]

Did the nice dog bite the child in the street in the street

Pedagogical use

The system may be used for demonstrating the linguistic structures, terminology and grammatical competence on different levels. It may also be used interactively as in the following questions and instructions to be presented by the computer:

Which categories are demonstrated in the sentence *Den snälla hunden var mycket snäll?* Which functional roles are demonstrated?

Try to draw the phrase structure tree and the functional tree for the sentence *Hunden som sprang bet det snälla barnet.*

Which level is demonstrated by the sentence *Barnet sade att hunden sprang*. Requires the answer level 5

Which level includes the word *troligen*? Correct answer: level 6.

What syntactic phenomena are available at level 4?

Which syntactic structures are not acceptable at level 3?

Which is the first level where questions with inverted word order is used?

What is the difference between functional representations and phrase/category representations?

Try to outline the processes that the system must apply in order to arrive at an English translation from a *Segram* functional (interlingua) representation.

Language acquisition

By associating *Segram* to the levels of Processability theory it is possible to simulate language acquisition. *Segram* can therefore give students insights in the language acquisition process. Traditionally, the acquisition of a language is generally seen as a gradual construction of the target grammar from no grammar to a native-like grammar. There is, however, a vast amount of empirical research on developmental levels in the acquisition of Swedish (for an overview, see Pienemann & Håkansson 1999), and these levels are used as a basis for *Segram*. *Segram* illustrates the constraints on each level in a clear and consistent way, without the variability which is inherent to natural learner language, and which sometimes makes it hard to see patterns. Since the lexicon of *Segram* is limited, the grammar can be focussed more clearly.

The grammatical model

Segram assumes (following Chomsky 1957) that it is possible to describe the form and meaning of Swedish sentences by a set of base (kernel) sentences plus a set of transformations. The base sentences of *Segram* are fixed sequences of noun phrases, verbs, adverb phrases, etc., and the meanings of the base sentences are represented by corresponding sequences of functional roles: subject, object, predicate, adverbials, etc., including word meanings. The base sentences of *Segram* have VS order, termed 'inverted' by Nordic-biased grammarians who regard the SV order as the default. The order in an intransitive base sentence in *Segram* is: finverb, subj, sentadv, negadv, manneradv, timeadv, placeadv.

The transformations apply to the base sentences in order to get different surface word orders, above all different initial constituents (in the fundament, to use Diderichsens term), as topic/focus. The sentence *Hunden sprang igår* 'The dog ran yesterday' thus starts from the base sentence corresponding to *sprang hunden igår*, from which *hunden* is preposed. In the sentence *Igår sprang hunden,* the time adverbial *igår* has been moved to the front. If nothing is moved to the front a question is generated: *Sprang hunden igår?* 'Did the dog run yesterday'.

The inverse basic word order has the advantage that only one movement is required: the movement to the initial position. The subject is already in the position after the finite verb in the base sentence and does not have to be moved from the front to this position.

This approach with underlying verb first is essentially the approach suggested by the Danish grammarian Paul Diderichsen (1946). In *Segram* the movements only concern the phrase structure representations not the functional representations, where the standard order of *Swetra* grammar (see Sigurd 1994) is used. The mode associated with the different transformations is registered.

The analysis of sentences by *Segram* is similarly done in two steps. If the sentence can be identified with one of the base sentences it is recognized as a yes/no question, if not it is identified as a declarative sentence. Imperative sentences are not covered by *Segram*. *Wh*-questions are not registered specially in the present version of *Segram.*

Sentences with a preposed adverbial immediately followed by the subject, and not by the verb, such as *Igår hunden sprang* pose a problem to our approach. They are accepted by some learners in the present version at level 3, but not later. Learning the inverse word order is regarded as an important

step in the acquisition of Swedish. In order to generate non-inverted word order *Segram* has to include a rule which first preposes the subject and then the adverb, as if the first operation was not enough. This rule occurs optionally at level 3, but not later. Alternatively, one may assume that these learners start with SV in the base clauses (as our English grammar), allow fronting transformations and reach level 4 by adding the rule V-movement to the second position later.

Programming *Segram* in Prolog

The grammar in *Segram* is programmed by generative arrow rules, DCG (Definite Clause Grammar). The sentence arrow rules have additional arguments on the left side of the arrow where the functional representation and the Swedish phrase structure representation to be displayed also as trees, are constructed. The grammar includes phrase and word rules where the word forms and their meanings are shown. The lexicon has to contain inflectional information as well. We will not give all details here.

Agreement in noun phrases is handled by a variable which has the same (or a compatible) value in all the constituents of the noun phrase. On the sentence level, a variable in the subject noun phrase must have the same value as the predicative adjective in Swedish predicative sentences in order to handle agreement. The following is a generative rule showing the base for intransitive verbs with one time adverbial. Items spelled with an initial capital letter are variables in Prolog.

```
s([subj(Nps),pred(Vi),tadvl(Tadv)],C,
[verbfras(Vs),subjnomfras(Npss),tidsadv(Tp)]) -->
{C=[vi(Vi,Vs),nps(Nps,Npss,Agr),tadvp(Tadv,Tp)]},nol(C). % intrns with time
adv
```

The first square bracket shows the functional representation, the C includes an operational representation which will surface as a string of words by the special predicate *nol*. The last parenthesis before the arrow includes labels which will appear in the Swedish phrase structure/word/phrase class diagram.

The following rule is a transformation which moves a time adverbial to the front.

```
s2(decl,tadv,F,C1,R1) --> {s(F,C,R,S,[]), % take a base sentence
C=[Vf,Ns|T],member(tadvp(Tadv,Ta),T),remove(tadvp(Tadv,Ta),T,T2),
C1=[tadvp(Tadv,Ta),Vf,Ns|T2], % remove time adv and put it first
R=[V1,N1|T1],remove(tidsadv(Ta),T1,Tb),
R1=[tidsadv(Ta),V1,N1|Tb]},
nol(C1). % e.g. Igår sprang hunden
```


Grammatical terminology

Swedish grammatical terminology is not completely standardized and some uncertainty is obvious among grammarians, teachers and students. *Segram* assumes two syntactic representations (trees), a phrase representation and a functional representation and so does the prestigeous grammar of the Swedish Academy (*SAG* 1999). The top node of the phrase representation is *sats* 'sentence' in *SAG* and the top node of the functional representation is named *mening* 'meaning'. *Swetra* uses *s* for both representations.

The grammatical terms developed since antiquity do not reflect a well established unified theory. It is clear that some terms reflect a dependency/ government/determination/modification/addition relation, where one member is the head of the other(s), which depend on the governor (head) and modify or determine it. Dependency oriented terms are (etymologically) *adverb* (adding to the verb), *adjective* (adding to a 'ject': subject or object), *attribute/ attributive* (attributing), *determiner* (determining the meaning), *modifier* (modifying the meaning). A determiner of an adjective is traditionally also called an *adverb* and so is a determiner of an adverb. *Segram* distinguishes between the adjective adverb *(adjadv)* and the adverb adverb *(advadv)*.

The phrase category structure terminology can be regarded as an extension of the traditional word class terminology in order to include multiword phrases. Subject and object noun phrases are distinguished in *Segram* in order to handle word order and the choice between *hon* 'she' and the object form *henne* 'her'.

The distinction between phrase and functional representations should ideally correspond to two sets of different terminology, but some terms are lacking in traditional grammar, e.g. functional terms for preposition (etymologically: before position), conjunction and subordinator. The functions of the sentence are traditionally: subject, predicate, direct object, indirect object, adverbial. The noun phrases have typically functions as subject and object and this is reflected in the terms subjnomfras, objnomfras.

The adverbial roles are typically played by adverbs, adverb phrases, prepositional phrases and certain subordinate clauses. The terms for functional roles are signaled by the suffix *-al* with adverbs: adverb – adverbial. A prepositional phrase may occur in a noun phrase as a post modifier *(postattribut)* as in *hunden på gatan* 'the dog in the street'. It is natural to call the heavier noun phrase the head *(h)* of the prepositional phrase, but one may alternatively argue for the preposition as the head. A similar problem is offered

by subordinate clauses where the subordinating conjunction or the relative pronoun may be regarded as the head.

We will not discuss these problems further but hope that the terms used in *Segram* should be acceptable or at least stimulate discussions of grammatical terminology. Students have long requested better grammatical terminology reflecting a unified system which is easier to understand.

We are grateful to Arthur Holmer, Lund for comments on this paper.

References

Chomsky, Noam. 1957. *Syntactic structures.* The Hague: Mouton.
Diderichsen, Paul. 1946. *Elementær dansk grammatik.* København: Gyldendal.
Lastow, Birgitta & Gisela Håkansson. 1997. 'Grammatical terminology and the application *Gramte*'. *Working papers* 46, 185-95. Dept of Linguistics, Lund University.
Pienemann, Manfred. 1998. *Language processing and second language development: processability theory.* Amsterdam: John Benjamins.
Pienemann, Manfred & Gisela Håkansson. 1999. 'A unified approach toward the development of Swedish as L2: a processability account'. *Studies in second language acquisition* 21, 383-410.
Sigurd, Bengt (ed.). 1994. *Computerized grammars for analysis and machine translation.* Lund: Lund University Press.
Svenska Akademiens grammatik (SAG). 1999. Stockholm: Norstedts.

Bengt Sigurd <Bengt.Sigurd@ling.lu.se>
Gisela Håkansson <Gisela.Hakansson@ling.lu.se>
Johan Dahl <Johan.Dahl@ling.lu.se>

Lund University, Dept. of Linguistics
Working Papers 51 (2005), 187–198

Machine translation of marine forecasts, quarterly company reports and recipes between Swedish, English, Malay and Chinese

Bengt Sigurd, Birgitta Lastow, Hong Gao and Mats Eeg-Olofsson

Abstract

This paper will report on four machine translation projects dealing with restricted domains and controlled or semi-controlled language. The computer program *Seatra* translates marine forecasts between Swedish and English. It is in daily use by the Swedish meteorological agency (SMHI) and generally needs no postediting. The same basic program but with different lexicons is used in the program *Maltra* which can translate official Malaysian marine forecasts into English and Chinese.
The program *Reptra* dealing with quarterly company reports and *Receptra* dealing with recipes use the same platform. All these programs (which can be used bidirectionally) will be presented and commented on, above all from a linguistic point of view. It is suggested that no more than 3 mistakes per 100 words be accepted if the postediting should not be too heavy, and this quality can be obtained in restricted-domain systems.

1 *Seatra*

The Canadian program *METEO* (Kittredge et al. 1973; Goldberg, Kittredge & Polguère 1988) is well known and the task to translate marine forecasts from Swedish into English suggested by the Swedish meteorological agency SMHI didn't seem too hard. Early work is described in Sigurd, Lastow & Vävargård 1996. The program *Seatra* was developed in Prolog (LPAProlog http://www.lpa.co.uk), using a technique which is based on lexicons containing single words, multi-word phrases, discontinuous phrases and some grammatical phrases such as noun phrases, prepositional phrases and adverbial phrases. The program is designed to produce simple and fast translation in a restricted domain without offering deep syntactic and