THE ORGANIZATION OF THE ROSETTA GRAMMARS

Jan Odijk

Philips Research Laboratories, P.O. Box 80.000

5600 JA Eindhoven, The Netherlands

ABSTRACT

In this paper the organization of the grammars in the Rosetta machine translation system is described and it is shown how this organization makes it possible to translate between words of different syntactic categories in a systematic way. It is also shown how the organization chosen makes it possible to translate 'small clauses' into full clauses and vice versa. The central concept worked out here in some detail is the concept of 'partial isomorphy' between subgrammars. The system as described here has been implemented and is currently being tested.

1 ROSETTA

In this section I will mention some essential properties of the Rosetta machine translation system. For more extensive descriptions of this system I refer to Landsbergen(1987), Appelo & Landsbergen(1986), Appelo, Fellinger & Landsbergen(1987) and Leermakers & Rous(1986).

The Rosetta machine translation system is an interlingual machine translation system based on Montague Grammar and developed at Philips Research Laboratories in Eindhoven for Dutch, English and Spanish. The grammars used in Rosetta are computational variants of Montague Grammar and are called M-grammars.

Rules in M-grammars, called M-rules, operate on syntactic surface trees. If a certain string *str* is associated with a syntactic tree of category C the notation $C(str)$ will be used.

An M-grammar in Rosetta is subdivided into a number of subgrammars. Each subgrammar is a rule package and is defined by specifying a *head* (the category of a designated argument from the import for the subgrammar), *export* (the categories of the syntactic trees yielded by the subgrammar)

and a *control expression* that indicates in which order M-rules must be applied.

In Rosetta3[1] a distinction is made between meaningful M-rules, called *rules* (that correspond to a meaning operation), and meaningless rules called *transformations* (that do not correspond to a meaning operation).

M-grammars are reversible. Because of this fact the translation relation between two sentences can be illustrated by showing their derivations in generation. The example derivations below will be restricted to generation only.

Within the Rosetta system the grammars of the relevant languages are attuned to one another in such a way that two sentences that are each other's translation can be derived starting from corresponding basic expressions by applying corresponding rules in the same way for both sentences. 'Corresponding' here means: 'corresponding to the same meaning operation or basic meaning', and it holds of meaningful M-rules only.

M-grammars attuned in this way are called *iso morphic grammars*, and therefore the method of translation used is called the *isomorphic grammar approach to machine translation*.

2 CATEGORIAL MISMATCHES

In order to derive simple sentences e.g. *Ziet hij de man?* one could design a grammar of Dutch that starting with a basic verb (*zie*) applies rules to derive the sentence mentioned. One could have rules combining a basic verb with arguments that it allows or requires (*hij, de man*) in a sentence, rules that determine the tense of this sentence (*present tense*) and rules that determine 'mood' and real

---

[1] The paper deals with the grammars in the third version of the Rosetta system, Rosetta3.

ize the sentence accordingly (in this case yes-no-question, main clause, realized by the order verb-subject-object), etc..

In order to derive the English sentence *Does he see the man?* as a translation of this Dutch sentence it would be required to design a grammar of English isomorphic to the grammar of Dutch sketched above. There must be a rule combining a verb (*see*) with its arguments (*he, the man*). There must be rules determining tense and aspect (*simple present tense* in the example sentence), and there must be rules that determine 'mood' and realize the sentence accordingly (yes-no- question, main clause, realized by introducing the auxiliary verb *do* and the order *do*-subject-verb-object), etc..

In this simple example the syntactic categories of the Dutch word *zien* and its translation into English *see* are the same. For these cases isomorphic grammars can be developed rather straight-forwardly.

However, machine translation systems must be able not only to translate between words of the same category, but also to translate between words of different syntactic categories. Some examples where a translation between words of different categories is required or desirable are given in the following table:

hij zwemt *graag* (Adv)
he *likes* to swim (Verb)
hij is *toevallig* ziek (Adv)
he *happens* to be ill (Verb)
het *mist* (Verb)
it is *foggy* (Adj)
hij *schaamt* zich ervoor (Verb)
He is *ashamed* of it (Adj)
hij is mij 3 gulden *schuldig* (Adj)
He *owes* me 3 guilders (Verb)
hij is in A'dam *woonachtig* (Adj)
He *resides* in Amsterdam (Verb)
Het is *voldoende* (Adj)
It *suffices* (Verb)

The Rosetta grammars must be able to deal with such cases in a systematic way.

It must also be possible to translate 'small clauses' into full clauses (finite or infinite) and vice versa. 'Small Clauses' are propositional units in which a subject-predicate relationship is expressed and which are not marked for tense and aspect (cf. Stowell (1981)). Some examples of full clauses and their corresponding 'small clauses' are given in the following table:

'Full Clauses'

Verb	He killed a man
Adj	He is intelligent
Prep	He is against the deal
Adv	He is here
Noun	He is a fool

'Small Clauses'

Verb	I had *the man killed*
Adj	I consider *him intelligent*
Prep	We got *him against the deal*
Adv	We got *him here*
Noun	They consider *him a fool*

Small clauses can occur only as complements within a clause. The fact that they are not marked for tense and aspect is reflected in the absence of auxiliary and copular verbs. If the grammar allows for translation of clauses into small clauses and vice-versa, then it becomes possible to derive e.g. *he seems ill* and *hij schijnt ziek te zijn* (lit. *he seems to be ill* as translations of each other. It is sometimes required to be able to translate small clauses into full clauses in some cases. The English sentence *he seems ill* cannot be translated into Dutch *hij schijnt ziek, which is ungrammatical, but must be translated into *hij schijnt ziek te zijn* (*he seems to be ill*) or into *het schijnt dat hij ziek is* (*it seems that he is ill*).

3 GRAMMAR ORGANIZATION

The global organization of the grammars (leaving transformations out of consideration) is the same for all languages, as is required by the isomorphic method.

The grammars are subdivided into five *projection subgrammars*[2], one for each major category (Verb, Noun, Prep, Adj, Adv).[3]

Each of these projection subgrammars consists of a number of subgrammars. The partitioning of projection subgrammars into subgrammars is motivated by the fact that it must be possible to use the output of each subgrammar as import to subgrammars of other projection subgrammars.

A schematic representation of the partitioning of a projection subgrammar into subgrammars is

---

[2]These are called *subgrammars* in Appelo, Fellinger & Landsbergen(1987)

[3]Apart from these subgrammars there is also a projection subgrammar to derive simple NPs, which is not partially isomorphic to the other projection subgrammars.

given in figure 1, where X is a variable over the major syntactic categories. A projection subgrammar is a set of subgrammars that define a projection for a given category X. A projection of some category X is a syntactic structure headed by X.

Each projection subgrammar is bifurcated as indicated in figure 1. If the rules and transformations in the XPPtoCLAUSE and CLAUSEtoSENTENCE subgrammars are applied a full clause is derived. If the rules and transformations of the XPPtoXPFORMULA (XPPtoXPF) and XPFORMULAtoXPP (XPFtoXPP) are applied a 'small clause' is derived.

The projection subgrammars are partially isomorphic to one another, which makes it possible to translate a subset of constructions from one projection subgrammar into constructions from some other projection subgrammar. Furthermore the XPPtoXPF subgrammars are partially isomorphic to the XPPtoCLAUSE subgrammars, and the XPFtoXPP subgrammars are partially isomorphic to the CLAUSEtoSENTENCE subgrammars. This makes it possible to translate a subset of full clauses into 'small clauses'.

The subgrammars are partially isomorphic to one another instead of fully isomorphic, because for certain full clauses no corresponding small clauses exist (e.g. there is no 'small clause' corresponding to the full clause *Is he ill?*).
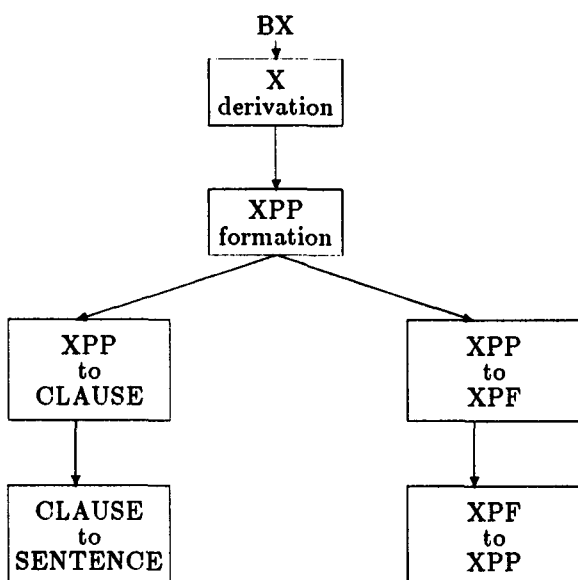

Figure 1: The global organization of the Rosetta projection subgrammars. *X* is a variable ranging over the major syntactic categories

The subgrammars indicated in figure 1 will be discussed in more detail now.

X-Derivation subgrammar deals with derivation and composition

XPPformation In this subgrammar the *argument structure* and the syntactic realization of arguments (whether they are subject, object, prepositional object, etc.) is accounted for. Voice (active, passive) is determined in this subgrammar.

XPPtoCLAUSE When this subgrammar is applied a full clause will be derived. In the subgrammar propositional complements and adverbials are introduced, tense and aspect is determined and auxiliaries are introduced accordingly. There are several transformations in this subgrammar to deal with the proper incorporation of propositional units into the structure, e.g. transformations dealing with control phenomena, and transformations dealing with the proper positioning of verbs (especially in Dutch). 'Hidden' arguments, e.g nonovert *by*-phrases in passives, are dealt with here.

XPPtoXPF When this subgrammar is applied a 'small clause' will be derived. The XPPtoXPF subgrammars are partially isomorphic to the XPPtoCLAUSE subgrammars, hence they contain similar corresponding rules. They contain rules introducing propositional complements, rules determining tense and aspect (which is considered to be present though dependent upon superordinate clauses) and rules dealing with hidden arguments.

CLAUSEtoSENTENCE In this subgrammar nonpropositional arguments are introduced and scope of quantificational expressions is accounted for. Furthermore the 'mood' of a sentence is determined, i.e. it is decided whether it is going to be a declarative, interrogative relative etc. clause.

XPFtoXPP This subgrammar is partially isomorphic to the CLAUSEtoSENTENCE subgrammars. It contains rules to introduce nonpropositional arguments and to account for scope. There are 'mood' rules determining whether the XPP is 'closed' (i.e. there are no free variables left in the structure), or 'open' (i.e. the subject variable is free).

4 ILLUSTRATIONS

The global organization of the grammar will be illustrated here by showing the derivation processes of certain examples involving a categorial mismatch and of some examples of translations from 'small clauses' into full clauses or vice versa. I repeat that only the generative part of the derivation need be demonstrated given the reversibility of the grammars. Furthermore only the correct derivation paths through the grammar will be illustrated, though in reality many false paths, i.e. paths through the grammar that do not lead to an actual sentence, are chosen by the grammar.

Since M-grammars are reversible it possible to 'translate' from e.g. Dutch into Dutch. In this way Rosetta functions as a paraphrase generator. Because of this, the partial isomorphy between two projection subgrammars can (and will) be illustrated by showing the parallel derivation of two paraphrases in generation.

In subsection 4.1 the derivation of *hij schaamt zich ervoor* and *he is ashamed of it* will be illustrated.

In subsection 4.2 the derivation of *I find him intelligent* and *I find that he is intelligent* will be illustrated.

In subsection 4.3 the derivation of *he seems intelligent, he seems to be intelligent* and *it seems that he is intelligent* as paraphrases of one another will be illustrated.

In subsection 4.4 the derivation of *de op haar verliefde man* and *de man die op haar verliefd is* as paraphrases of one another will be illustrated.

In subsection 4.5 the derivation of *hij zwemt graag* and *he likes to swim* will be illustrated.

4.1 Adjective-Verb

The derivations of *hij schaamt zich ervoor* (lit. *he ashamed himself therefor*) and its translation *he is ashamed of it* run as follows. The verb *schamen* is a two place verb so that it can be combined with two variables, x_1 and x_2, in the VPPformation subgrammar yielding VPP(x_1 x_2 *schaam*). In the English grammar, the adjective *ashamed* is also combined with two variables in the ADJPPformation subgrammar, yielding ADJPP(x_1 *ashamed* x_2). The Dutch verb *schamen* must realize its second argument as a prepositional object with the preposition *voor*. The English adjective *ashamed* must realize its second argument as a prepositional object as well, though headed by the preposition *of*. Pattern transformations are applied to yield this

effect: VPP(x_1 *voor* x_2 *schaam*) and ADJPP(x_1 *ashamed of* x_2).

In the VPPformation subgrammar a Voice-rule applies determining that the structure is in active voice. In the ADJPPformation subgrammar a corresponding rule applies which has no visible effects.

In Dutch a transformation spelling out a reflexive pronoun must apply, since the verb *schamen* is a so-called inherently reflexive verb. This yields VPP(x_1 *zich voor* x_2 *schaam*). Since spelling out these reflexive pronouns is achieved by transformations, no corresponding M-rule need apply in English.

These structures are both input to the XPPto-CLAUSE subgrammar where they are turned into clauses. In Dutch this yields CLAUSE(x_1 *zich voor* x_2 *schaam*). In English the copula *be* is introduced, yielding CLAUSE(x_1 *be ashamed of* x_2). Tense and aspect rules are applied in both cases, putting the structures in present tense, yielding CLAUSE(x_1 *zich voor* x_2 *schaamt*) and CLAUSE(x_1 *is ashamed of* x_2).

Substitution rules substitute the NPs *het* and *it* respectively for the variables x_2 and the NPs *hij* and *he* respectively for the variables x_1. This yields in English CLAUSE(*he is ashamed of it*) and in Dutch CLAUSE(*hij zich voor het schaamt*). An obligatory transformation turns *voor het* in Dutch into *ervoor*.

Mood rules are applied in Dutch and in English. The clauses are turned into declarative main clauses. In English this yields SENTENCE(*he is ashamed of it*) and in Dutch this yields SENTENCE(hij zich ervoor schaamt). Application of the transformation putting the finite verb in 'second' position and application of an obligatory topicalization transformation in Dutch yield the actual sentence *Hij schaamt zich ervoor*.

4.2 Adjective-declarative clause with main verb

The parallel derivations of *I find him intelligent* and *I find that he is intelligent* run as follows. In the ADJPPformation subgrammar the adjective *intelligent* is combined with a variable x_1 yielding ADJPP(x_1 *intelligent*) This ADJPP can be turned either into a clause yielding CLAUSE(x_1 *be intelligent*) (in the XPPtoCLAUSE subgrammar) or it can continue as an adjectival construction (in the ADJPPtoADJPF subgrammar). Tense and Aspect rules determine that the tense of this construction is dependent on the tense of a superordinate construction yet to be formed. A substitu-

tion rule substitutes the NP containing *he* for x_1 yielding ADJPF(*he intelligent*) and CLAUSE(*he is/was intelligent*) respectively. Finally Mood rules determine the way the clause and the ADJPF are realized. The mood rule forming subordinate declarative clauses is mapped onto the mood rule forming 'closed' ADJPPs (CADJPP). Applying these mood rules yields SENTENCE(*that he is/was intelligent*) and CADJPP(*he intelligent*). These results can be used later in the derivation as arguments to the verb *find*.

To derive the rest of the structures the verb *find* is combined with two variables (x_2, x_3) yielding VPP(x_2 *find* x_3). The substructures built earlier can be substituted for x_3 yielding VPP(x_2 *find he intelligent*) and VPP(x_2 *find that he is/was intelligent*) respectively. After turning these VPPs into clauses and applying tense and aspect rules (putting the sentences in present tense) the tense of the subordinate clause can be determined. This yields CLAUSE(x_2 *find he intelligent*) and CLAUSE(x_2 *find that he is intelligent*). Applying a case-assignment transformation and substituting the NP(*I*) for x_2 yields CLAUSE(*I find him intelligent*) and CLAUSE(*I find that he is intelligent*) respectively. Applying a mood rule that turns these clauses into declarative main clauses yields the actual sentences.

4.3 Adjective-declarative clause with copula

The derivations of *he seems intelligent, he seems to be intelligent* and *it seems that he is intelligent* starts in the same way as in the preceding section. A CADJPP(*he intelligent*) and a SENTENCE(*that he is/was intelligent*) are generated. In addition to the mood rule forming finite declarative subordinate sentences a mood rule forming 'closed' infinite declarative subordinate sentences containing *to* can be applied. This rule forms the SENTENCE(*he to be intelligent*) out of the CLAUSE(*he be intelligent*).

These results can be used as arguments to the verb *seem*. In the VPPformation subgrammar a VPP(*seem* x_2) is formed from the verb *seem* and the variable x_2. The SENTENCEs and the CADJPP obtained earlier are substituted for the variable x_2. This yields the structures VPP(*seem he intelligent*), VPP(*seem that he is/was intelligent*) and VPP(*seem he to be intelligent*) respectively.

A transformation turns the subject of embedded infinitival complements of verbs such as *seem* into the subject of *seem*, and the NP(*it*) is inserted as

a subject of the verb *seem* if it has a finite complement. After determining tense and aspect and applying a subject-verb agreement transformation this yields the structures CLAUSE(*he seems intelligent*), CLAUSE(*it seems that he is intelligent*) and CLAUSE(*he seems to be intelligent*) respectively. Applying the mood rule forming declarative main clauses yields the actual sentences.

In the grammar of Dutch parallel derivations can be made. The CADJPP(*hij intelligent*) (cf. *he intelligent*) and the SENTENCEs *dat hij intelligent is/was* and *hij intelligent te zijn* (cf. *that he is/was intelligent* and *he to be intelligent* resp.) can be derived.

Some of these results can be used as arguments to the verb *schijnen* '*seem*'. In the VPPformation subgrammar the verb *schijnen* is combined with the variable x_2 into a VPP($x2$ *schijn*). The SENTENCEs obtained earlier can be substituted for this variable, but the CADJPP(*hij intelligent*) cannot, because the Dutch verb *schijnen* does not take CADJPPs as a complement.

The derivation continues with the two results VPP(*dat hij is/was intelligent schijn*) and VPP(*hij intelligent te zijn schijn*). Application of several transformations to deal adequately with such complements in Dutch (Verb-raising and extraposition (in the sense of Evers(1975)), subject-to-subject-raising and some others) and application of tense and aspect rules yields CLAUSE(*het schijnt dat hij intelligent is*) and CLAUSE(*hij intelligent schijnt te zijn*) respectively.

Application of mood rules forming declarative main clauses and some obligatory transformations in Dutch yields the actual sentences *Het schijnt dat hij intelligent is* and *Hij schijnt intelligent te zijn* as translations of the English sentences derived earlier.

4.4 Adjective-relative clause

The derivations of *(de) op haar verliefde (man)* (*the man in love with her*) and *(de man) die op haar verliefd is* (*the man that is in love with her*) runs as follows.

The adjective *verliefd* '*in love*' is a two-place adjective that is combined with two variables x_1 and x_2 in the ADJPPformation subgrammar. This yields a structure of the form ADJPP(x_1 *verliefd* x_2). The adjective *verliefd* must realize its second argument (x_2) as a prepositional object that can occur in front of the adjective. Pattern transformations introduce the preposition required, yielding ADJPP(x_1 *op* x_2 *verliefd*).

This ADJPP can be turned into a clause, or it can be turned into an adjectival phrase. The XP-PROPtoCLAUSE subgrammar changes the structure mentioned in the following way: CLAUSE(x_1 op x_2 verliefd zijn), where the top category has been turned into CLAUSE and the copula zijn 'be' has been introduced. To form an adjectival phrase the ADJPP is input to the XPPtoXPF subgrammar, yielding ADJPF(x_1 op x_2 verliefd). In both subgrammars tense and aspect rules apply.

In the CLAUSEtoSENTENCE subgrammar a transformation is applicable that moves the subject variable x_1 into a position where normally relative pronouns would appear. In this particular structure this has no effects on the left-right order, but the relation that x_1 bears is changed.

In the CLAUSEtoSENTENCE subgrammar and in the ADJPFtoADJPP subgrammar the NP(zij) 'she' is substituted for variable x_2 and the appropriate Case form (accusative) is assigned to it. This yields CLAUSE(x_1 op haar verliefd is/was) and ADJPP(x_1 op haar verliefd).

Finally Mood rules turn the ADJPP into an 'open' ADJPP (OADJPP) yielding OADJPP(x_1 op haar verliefd), and they turn the CLAUSE into a relative subordinate clause: SENTENCE(x_1 op haar verliefd is). These structures can be used by rules in the NP-subgrammar that introduce these structures as modifiers and bind variable x_1.

4.5 Graag-like

In the Dutch sentence hij zwemt graag the adverb graag appears. This adverb must be mapped onto the English verb like in the translation he likes to swim.

It is assumed that the Dutch adverb graag is a two place-function. This is required in Rosetta, because its translation like is a two-place function[4]. However, apart from being required in Rosetta, it is also plausible for independent reasons that graag is a two-place function: the adverb graag imposes selectional restrictions upon the subject of the sentence it appears in (cf. ?het regent graag or ?de steen valt graag, which are as odd as their English counterparts ?it likes to rain and ?The stone likes to fall). If we assume that predicates impose selectional restrictions only upon their arguments, then it must be the case that the subject of the sentence is an argument of graag, or that the argument of

---

[4]This contrasts with the approach in the Eurotra framework, where this requirement does not hold. See Arnold et. al. (1988)

graag is coindexed with the subject of the sentence. I will assume the latter.

Starting with the subordinate infinitival clause in English, we combine the verb swim, which takes one argument, with a variable x_1 as its subject, yielding: VPP(x_1 swim) Similarly in the Dutch grammar the translation of swim, zwem, is combined with x_1: VPP(x_1 zwem) In the VPPformation subgrammar the voice rule to form active clauses is applied.

After the VPPformation subgrammar the derivation continues in the XPPtoCLAUSE subgrammar. An M-rule to make the sentence infinitival is applied. Corresponding rules are applied in the XPPtoCLAUSE subgrammar of Dutch.

In the CLAUSEtoSENTENCE subgrammar no arguments are substituted. In English a rule is applied that makes the clause an infinitival subordinate clause containing to and containing a free variable in subject position which will later be subject to control transformations. This yields a structure of the form SENTENCE(x_1 to swim)

In Dutch a corresponding rule is applied that makes the clause an infinitival subordinate clause without te with a free variable in subject position. This yields the following structure in Dutch: SENTENCE(x_1 zwemmen) These structures will be used later on.

In the VPPformation subgrammar of English the two-place verb like is combined with two variables, x_1 and x_2. This yields: VPP(x_1 like x_2) The voice rule to form active sentences is applied.

Correspondingly, in the ADVPPformation subgrammar of Dutch the adverb graag is combined with two variables, x_1 and x_2, and a voice rules is applied. This yields: ADVPP(x_1 graag x_2)

In the English subgrammar XPPtoCLAUSE the sentential structure derived above is substituted for the variable x_2, yielding: VPP(x_1 like [x_1 to swim])

A control transformation deletes the second occurrence of x_1. Tense and aspect rules apply which turn the structure into a finite clause in present tense.

In the Dutch subgrammar XPPtoCLAUSE the sentential structure derived above is substituted for the variable x_2 by a special rule that takes care of substitution of sentential complements into ADVPPs. This special rule deletes the ADVPP node and replaces it by a VPP node, turns the ADVP into a modifier inside this VPP, makes the VP of SENT the VP of this VPP, deletes the variable x_1 inside SENT and un-does the morphological effects of tense rules. This yields the structure: VPP(x_1

ADVP(graag) zwem)

These structures continue their normal derivation. Tense and aspect rules apply, the NP argument *hij (he)* is substituted for x_1 and the sentence is made into a declarative main clause, yielding in English: SENTENCE(*he likes to swim*) and in Dutch, after application of the transformation of 'Verb second' and an obligatory topicalization transformation: SENTENCE(*hij zwemt graag*)

5 CONCLUSION

It has been shown that the concept of partial isomorphy between subgrammars makes it possible to translate between words of different syntactic categories and between 'small clauses' and full clauses in a systematic way. Furthermore, it has been shown that one of the most difficult cases of translation between words of different categories, viz. the *graag/ like* translation problem, can be reduced to having only one special rule, given partial isomorphy between subgrammars.

ACKNOWLEDGEMENTS

REFERENCES

Appelo, L. , C. Fellinger and J. Landsbergen (1987), 'Subgrammars, Rule Classes and Control in the Rosetta Translation System', Philips Research M.S. 14.131, to appear in: *Proceedings of 3rd Conference ACL*, European Chapter.

Appelo, L. and J. Landsbergen (1986), 'The Machine Translation Project Rosetta', Philips Research M.S. 13.801, In: *Proceedings First International Conference on the State of the Art in Machine Translation*, Saarbrücken, pp. 34-51.

Arnold, D., S. Krauwer, L. des Tombe & L. Sadler (1988), "Relaxed' Compositionality in Machine Translation', in: *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Carnegie Mellon University, Center for Machine Translation, Pittsburgh, Pennsylvania.

Evers, A. (1975), *The Transformational Cycle in Dutch and German*, diss. University of Utrecht.

Landsbergen, J. (1987), 'Isomorphic grammars and their use in the Rosetta Translation System', Philips Research M.S. 12.950, Paper presented at the Tutorial on Machine Translation, Lugano, 1984, in: M. King (ed.), *Machine Translation, the State of the Art*, Edinburg University Press.

Leermakers, R. and J. Rous (1986), 'The Translation Method of Rosetta', Philips Research M.S. 13.701, In: *Computers and Translation*, Vol 1, number 3, pp. 169-183.

Stowell, T. (1981), *Origins of Phrase Structure*, Ph. D. dissertation, MIT.