16
Rosetta

The MT project at the Philips Research Laboratories in Eindhoven (Netherlands) is one of the most innovative experimental systems at the present time. Its essential feature is the attempt to devise interlingual representations based on the principles of Montague grammar, a theory which directly links syntax and semantics. Important theses are being explored: the reversibility of grammars, the compositionality of meaning, and the potential isomorphism of grammars.

16.1 Background

The project has its roots in earlier research at Philips on a question-answering system, PHLIQA. The task was to convert a question expressed in English into the logical representation language of the database. It was undertaken by a parser based on a context-free grammar where every grammar rule was coupled to a translation rule into the logical language. In other words, the logical interpretation of the question was based on the structural relations among its elements. However, the translation was not direct: the context-free representations were transformed first into a hybrid 'logical'-cum-'deep' syntactic structure before the truly logical representation was obtained. The unsatisfactory nature of this hybrid approach led to the design of a new grammar which would be fully compositional and in which the rules were more powerful than those of context-free grammars. It was concluded that the grammars described by the philosopher Richard Montague offered an attractive model for this approach.

Jan Landsbergen decided to explore the possibilities in the Rosetta project, which began in 1980. Initially two small experimental systems were built: Rosetta1

and Rosetta2. A larger project began in 1985, to be in two phases. The first phase has concentrated on the essential linguistic and computational framework and the building of a research system (Rosetta3) for the translation of short simple sentences from Dutch into English and Spanish and from English or Spanish into Dutch. Dictionaries are small and the system generates all possible translations. No corpus of actual texts has been tested. The second phase, which began in 1989, is devoted to the development of a more robust version of Rosetta3 and then the construction of a prototype system for a real application (Rosetta4). The eventual aim is a system for users not knowing target languages; it is to include monolingual interactive disambiguation during analysis, and to produce output not requiring post-editing. These practical requirements have not yet been addressed. All research has concentrated on the theoretical and linguistic foundations.

16.2 Montague grammar

The main characteristic of Montague grammar is the binding of semantic interpretations to structural relations. Montague grammars obey the principle of **compositionality**, namely that the meaning of an expression is a function of the meaning of its parts. Since the parts are defined by the syntax, there is a close relation between syntax and semantics.

A Montague grammar specifies a set of 'basic expressions' and a set of syntactic rules. The basic expressions are the smallest meaningful units and the rules prescribe how larger expressions (and ultimately sentences) can be constructed from these basic expressions. The rules are applied bottom-up.

We may illustrate with a simple example grammar, containing two basic expressions *car* and *pass*, and two rules (1a,b) for English.

(1a) ER1 = a rule applied to a noun which produces an indefinite singular noun phrase (by adding an article *a*).

(1b) ER2 = a rule applied to a noun phrase and an intransitive verb which produces a sentence, with the noun phrase as subject, in the past tense.

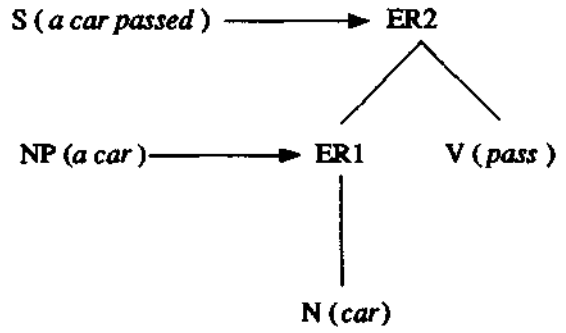
From these two basic expressions and these two rules can be generated (2).

(2) A car passed.

The process of deriving the sentence can be made explicit in a **syntactic derivation tree** (3).

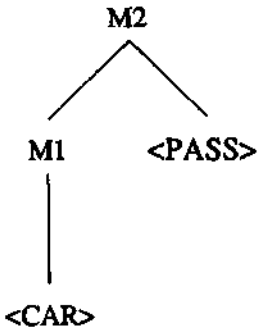
The semantic component of a Montague grammar assigns a semantic interpretation to an expression by relating it to the semantic domain of a 'possible world'. Semantic values can be assigned either directly or indirectly. In direct interpretation (the method adopted in the Rosetta project), each basic expression is associated directly with an 'object' in the domain (e.g. an individual) and each rule is associated with an operation on objects in the domain (e.g. a function). There are thus 'basic meanings' corresponding to basic expressions and 'meaning operations' corresponding to syntactic rules. The semantic value (meaning) of an expression is thus defined with the help of the syntactic derivation tree, i.e. in parallel with the application of the syntactic rules the meaning operations corresponding to these

(3)



rules are applied to the meanings of their arguments (constituents), starting with the basic expressions and again working bottom-up. The process of deriving the meaning parallels the process of deriving a syntactic representation. It can be represented in a **semantic derivation tree** which has the same geometry as its corresponding syntactic derivation tree but is labelled with the names of basic meanings and meaning operations (or 'rule meanings'). The semantic derivation tree for (3) would be as in (4), where the basic meanings <CAR> (= the property 'being a car') and <PASS> (= the property 'passing') correspond to the basic expressions N(*car*) and V(*pass*), and rule meanings M1 and M2 correspond to syntactic rules R1 and R2. The semantic well-formedness of a sentence is thus determined by the truth-values of the meaning rules which have been applied.

(4)



It is evident that from semantic derivation trees may be derived logical expressions, and this is frequently the preferred option by Montague grammarians (employing intensional logic formalisms, which can be interpreted with respect to a model of 'possible world semantics', i.e. in a model-theoretic semantics.) One possibility in an MT system would be to use these logical expressions as interlingua representations. However, it is argued that this would entail the loss of information about the surface forms of sentences (texts), and this information can be vital for producing satisfactory translations. In addition, there would be the difficulty of devising a single logical formalism for a wide variety of languages. Consequently, the Rosetta project has taken a different approach: it uses the semantic derivation trees themselves as interlingua representations. The logical

interpretation of sentences is not considered to be necessary for the purposes of translation; the semantic derivation tree contains exactly the relevant information that has to be preserved during translation.

16.3 Reversibility and isomorphism

As well as the principles of compositionality and interlinguality, the Rosetta grammars conform to three other principles. The first is **explicitness**: not only are the grammars of both source and target languages to be defined independently but all linguistic and translation processes are precisely expressed in the grammars. No procedures are to be implicit in the programming implementation.

The second is the **one grammar** principle: the same grammar is used for generation and for analysis of sentences, i.e. the grammars are intended to be reversible. The most important requirement is the reversibility of the syntactic rules. For example, the reverse rules for the example grammar in (1) above would be as in (5).

(5a) ER′1: a rule applied to an expression of the form NP (α) which produces an expression of the form N (α)

(5b) ER′2: a rule applied to an expression of the form S (α *βed*) which produces two expressions, one of the form NP (α), the other of the form V (β).

If ER′2 is applied to the sentence S (*a car passed*), the result is the pair NP (*a car*) and V (*pass*), where the latter is a basic expression. The application of ER′1 to NP (*a car*) yields the basic expression N (*car*). Analysis is thus successful if it reduces a sentence to its basic expressions. The analysis process itself is made explicit in a derivation tree, which is identical to (3).

The third principle, **isomorphism**, follows from the decision to adopt the compositionality principle and makes possible the use of semantic derivation trees as interlingual representations. Two sentences are considered translations of each other if they have the same semantic derivation trees, and hence corresponding syntactic derivation trees. As a consequence, the grammars of two languages have to be attuned to each other, so that for each basic expression in one grammar there is at least one corresponding basic expression in the other with the same meaning, and so that for each rule in one grammar there is at least one corresponding rule in the other. For example, a Dutch grammar which is isomorphic to the English grammar in (1) and (5) would contain two basic expressions N (*auto*) and V (*passeer*) with the meanings <CAR> and <PASS> respectively, thus corresponding to N (*car*) and V (*pass*) in the English grammar, and two rules (6a,b)

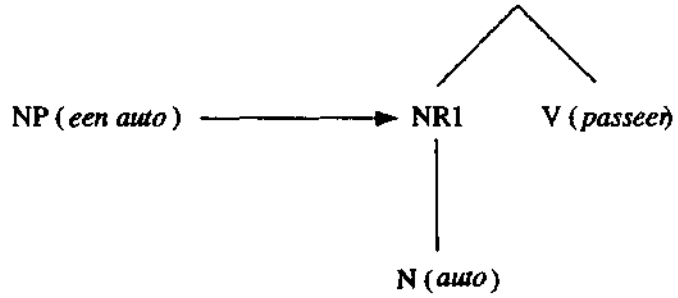
(6a) NR1: a rule applied to an expression of the form N (α) which produces an indefinite noun phrase of the form NP (*een* α)

(6b) NR2: a rule applied to a noun phrase of the form NP (*een* α) and an intransitive verb of the form V (β) which produces a sentence of the form S (*er* β*de een* α)

The application of these rules to the two basic expressions would generate (7) with the syntactic derivation tree (8).

(7) *Er passeerde een auto.*

(8) S (*er passeerde een auto*) ⟶ NR2



This tree would have the same semantic derivation tree as (4) thanks to the correspondences between the Dutch rule NR1 and the meaning rule M1, between NR2 and M2, between *auto* and the basic meaning <CAR> and between *passeer* and <PASS>. As a consequence, the Dutch and English grammars would be isomorphic, with corresponding basic expressions (*auto* and *car*, *passeer* and *pass*) and corresponding syntactic rules (NR1 and ER1, NR2 and ER2).

It is stressed by Rosetta researchers that isomorphism and not interlinguality is the primary characteristic of the framework. The essential condition for two sentences being translations of each other is, therefore, that they have isomorphic syntactic derivation trees: they have corresponding sets of rules, relating to the same meaning rule, and corresponding sets of basic expressions, relating to the same basic meaning. Not only are the grammars of the languages of the system designed in parallel and for the specific purpose of translation but the interlingua is also constructed specifically for the particular languages concerned. The isomorphism principle expresses the essence of the Rosetta compositional theory of translation.

16.4 Translation processes

We can now describe the Rosetta translation processes. The system has eight stages as illustrated in Figure 16.1.

The Rosetta grammars, called 'M-grammars' showing their affinity to Montague grammars, have three basic components each of which is reversible: a morphology component, a syntactic component and a semantic component. The syntactic component is divided into a part which deals with surface syntactic structures and a part which mediates between these structures and syntactic derivation trees. The semantic component deals with the interfaces between syntactic derivation trees and semantic derivation trees.

The stages are illustrated by working through the translation of the English interrogative sentence (9) into its Dutch equivalent (10).

(9) Does he love flowers?

(10) *Houdt hij van bloemen?*

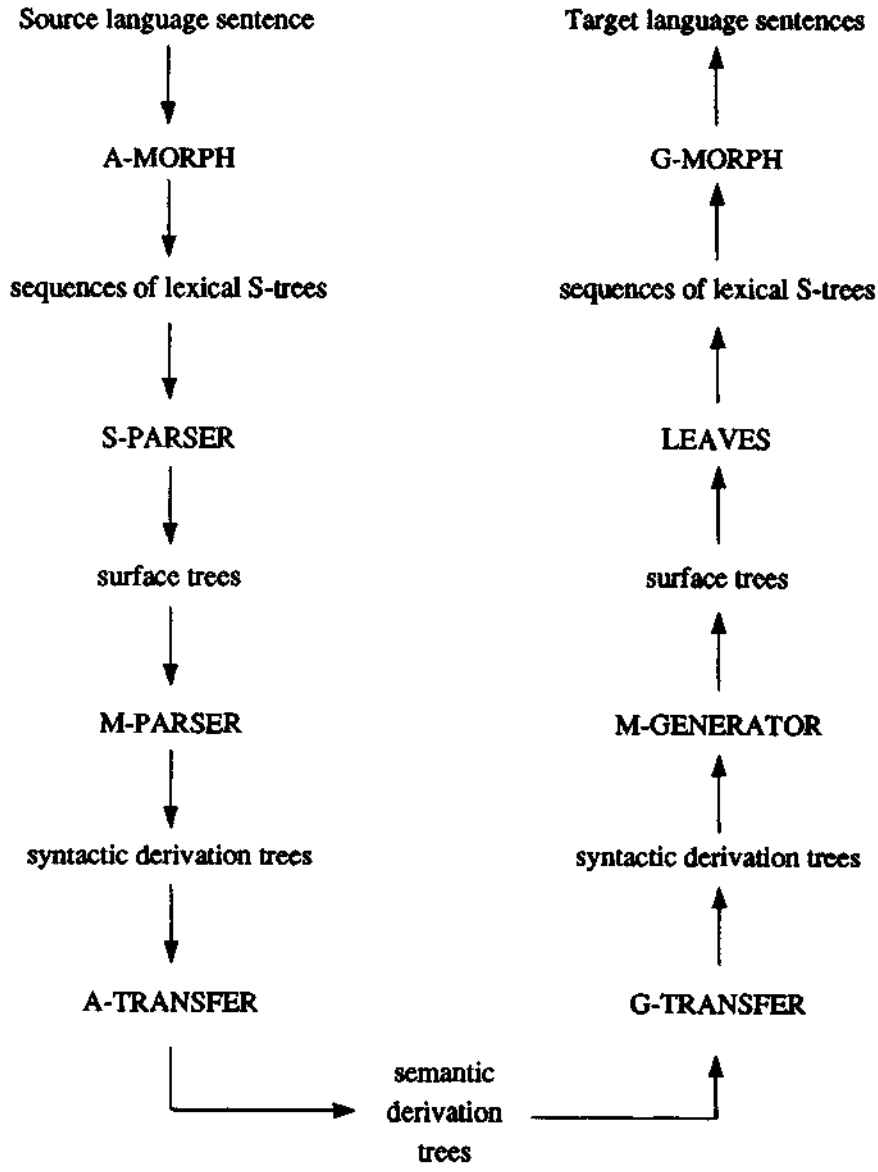
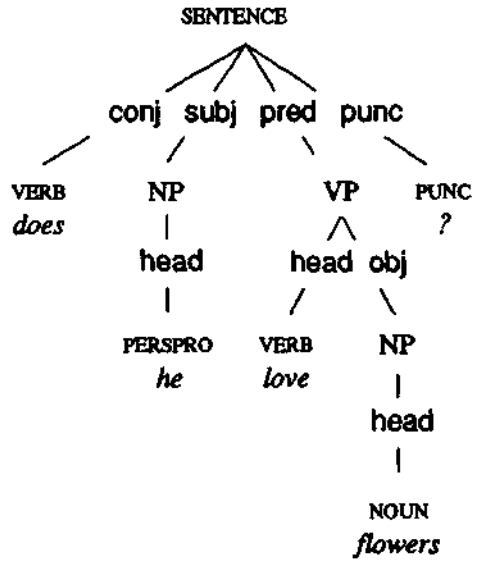


Figure 16.1 Rosetta translation process

In the first stage, **morphological analysis** (A-MORPH), input strings are decomposed into stems (*do, love, flower*) and affixes (*-es, -s*) to produce sequences of 'lexical S-trees'. Whereas in Montague grammar basic expressions are simple lexical items, in Rosetta they are defined as ordered trees ('S-trees' = surface trees) comprising one or more labelled elements. They may therefore be 'idiomatic' expressions of several lexical items in particular relationships, as we shall see below (section 16.8.)

The second stage is the first part of a two-step **syntactic analysis**, in which the S-PARSER produces sets of tentative analyses. Categorial homography is resolved (e.g. *love* is here not a noun but a verb) but no lexical or syntactic ambiguity. For sentence (9) the resultant surface tree is as in (11).

(11)

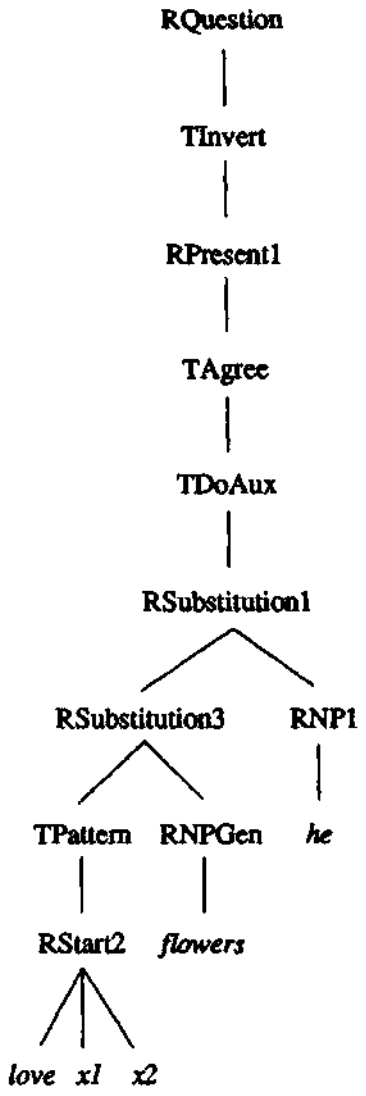


In the second part of syntactic analysis, the M-PARSER selects the syntactically correct tree structures and lays the foundation for representing the meaning of the sentence, i.e. it produces a syntactic derivation tree (in the case of (11) the tree illustrated in (12) below). The M-PARSER includes rules which may change nodes and relations in trees or modify the attributes of specific nodes. A distinction is made between rules and transformations. **Rules** (in the strict sense) are operations which have meaning or convey information relevant for translation; **transformations** are 'meaningless' and serve only to adjust structures for a particular language, they are language-specific operations which do not convey translationally relevant information. Both types are applied top-down to input surface trees under the control of the M-PARSER.

First, the sentence as a whole is considered: in the case of (11) an interrogative structure is recognised and consequently a rule RQuestion is applied which removes the question mark and which now permits the operation of a transformation TInvert to produce a statement (*He does love flowers*). Note that the inversion transformation itself is not relevant for translation (it is language-specific), only the identification of the question status is required.

Next, the identification of the tense involves the application of the rule RPresent1 (i.e. in this case 'present'), which requires also a check that subject and verb agree in person and number, a requirement specific to English, which is accomplished by the transformation TAgree (*He do love flowers*). It is followed by the recognition that the auxiliary *do* has been used, another 'meaningless'

(12)



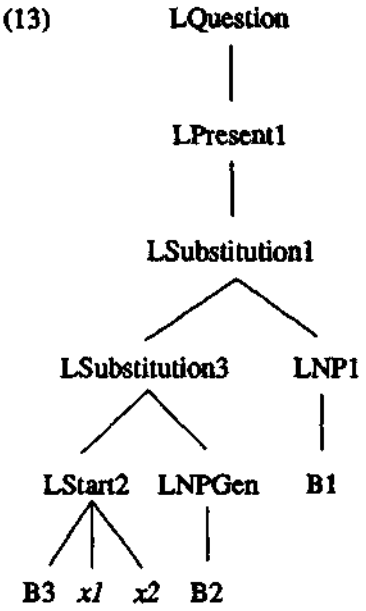
element in the context of English question structures, and so it is removed by the transformation TDoAux. We now have the basic elements *He love flowers*.

The subject and object are now replaced by abstract variables; this is done by two substitution rules (RSubstitution1 and RSubstitution3). The two extracted noun phrases are further reduced by noun phrase rules: the first RNP1 applies to noun phrases consisting of bare person pronouns, the second RNPGen applies to generic noun phrases (e.g. in English, nouns without articles). The final stage checks that the remaining structure (a verb with two abstract variables as subject and object) is acceptable by confirming the valency pattern for this specific verb, i.e. a language-specific operation and thus a transformation. The rule TPattern confirms that *love* can have a direct object, and renames the relation 'obj' by the

more neutral term 'argument'. Finally, RStart2 checks that the verb can have a subject and one other argument. The result of all these operations is the syntactic derivation tree (12).

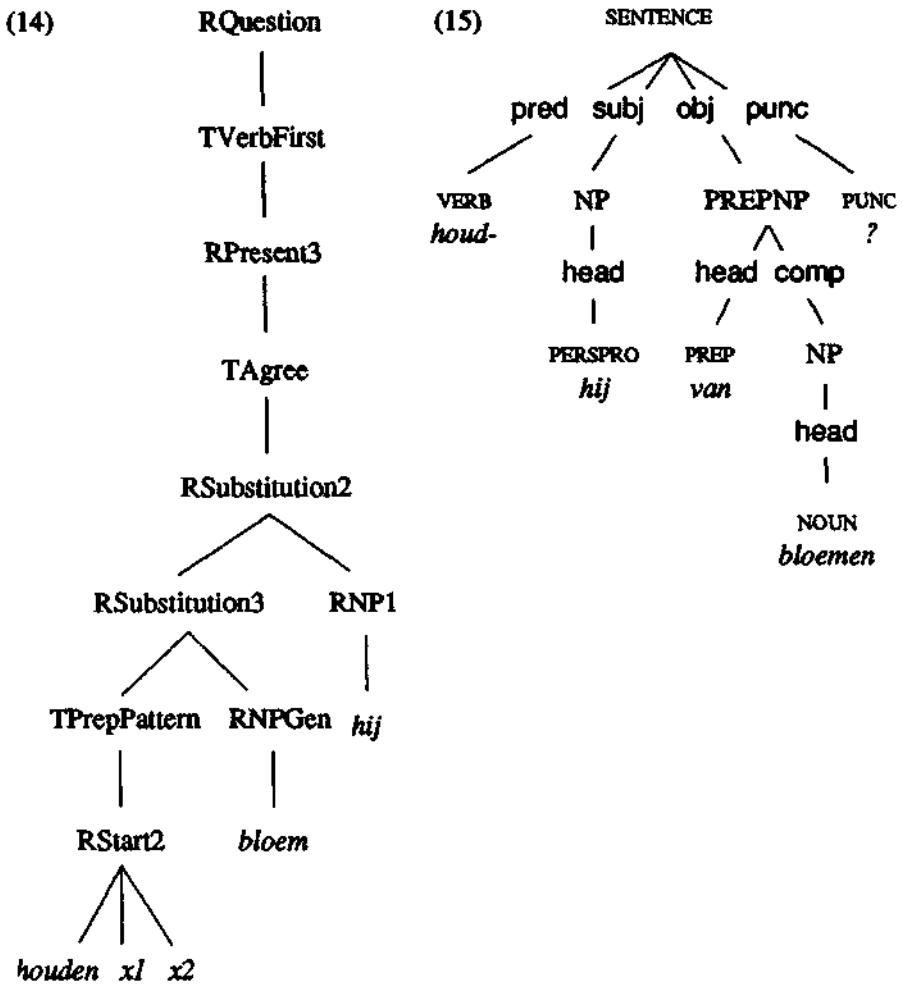
The syntactic derivation tree (12) is used to determine the meaning representation, a semantic derivation tree (13), by mapping each ('meaningful') rule onto a corresponding 'meaning operation' of the interlingua and by substituting interlingual basic expressions for source language basic expressions. This is performed by the next stage, **analytic transfer** (A-TRANSFER). Transformations in the syntactic derivation tree have no representation since they are language-specific; thus certain function words, such as the auxiliary *do*, have no corresponding basic expressions in the interlingual representation.

(13)



This semantic derivation tree is the source for a syntactic derivation tree in Dutch (14), which in conformity with the principle of isomorphism has the same meaning as the English tree (12). It is produced by the **generative transfer** (G-TRANSFER) module.

Generative transfer operates bottom-up producing a number of potential trees; only during the application of M-GENERATOR can the particular one for (13) be identified. In other words, M-GENERATOR has two functions (similar to those of M-PARSER): to validate syntactic derivation trees and select the correct ones, and to convert such trees into surface structures. (Obviously there can always be more than one output for a given meaning representation, see also 16.6 below.) It should be noted that the selection of *houden* as translation of basic expression B3 (i.e. *love*) entails a different pattern type in Dutch, one in which the object is preceded by a preposition (TPrepPattern). A further language-specific feature in (14) is



the placing of the verb at the beginning of Dutch questions (transformation rule TVerbFirst). The result of M-GENERATOR might be the surface structure (15).

The final generation stages are the production of a sequence of lexical S-trees by the LEAVES module, which picks off the words from the surface tree (15), and then the generation of correct morphological forms (G-MORPH), which in this case would determine the spelling of *houden* in the third person (i.e. *houdt*).

16.5 Structural correspondences

Since Rosetta does not preserve syntactic structures, it is able to change grammatical categories and to produce target structures which differ markedly from source structures. As an example of category change consider the generic noun rule (RNPGen) above. In English (and Dutch) the genericity of objects is expressed frequently without an article (*flowers*); in Spanish, however, the rule for

generic nouns would have to trigger the production of a definite article (*las flores*). An example of structure change has already been given: while *love* has a direct object, *houden* requires a prepositional object (*houden van*).

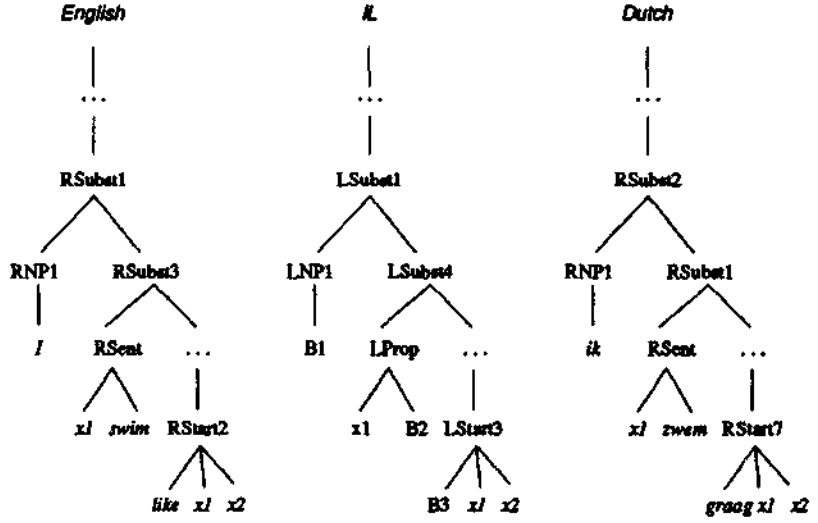
More complex relations are exhibited by correspondences such as those between Dutch sentences containing the adverb *graag* and English sentences containing *like*, as in (16), similar to the *like/gern* examples seen in Chapter 6.

(16a) *Ik zwem graag.*

(16b) I like to swim.

In Rosetta, the isomorphism of the two grammars is maintained by proposing syntactic and semantic derivation trees as in (17).

(17)



The English rules are quite straightforward: RSubst3 replaces a predicate with an infinitive (*like to swim*) by a sentential complement (*x1 swim*) and a variable (*like x1 x2*). These are mapped directly onto the interlingual rules. On the Dutch side there are a number of possible correspondences for LSubst4, one of which (RSubst1) takes a propositional structure with *graag* as its main element and by a series of transformation rules places it as an adverbial modifier of *x1 zwem*. It is assumed that *graag* is a two-place function. Apart from the motivation to maintain isomorphy with the English two-place function *like*, it is argued that the adverb *graag* imposes selection restrictions on the subjects of sentences which parallel those for *like*. Just as (18a) and (19a) are odd, the Dutch equivalents (18b) and (19b) are equally odd.

(18a) *It likes to rain.

(18b) **Het regent graag.*

(19a) *The stone likes to fall.

(19b) **De steen valt graag.*

In other words the conditions attached to the variable *x1* in Dutch correspond to those attached to the variable *x1* in English.

16.6 Subgrammars

Problems of categorial difference which this example demonstrates are handled in Rosetta by dividing M-grammars into five 'projection subgrammars', one for each major category (verb, noun, preposition, adjective, adverb). Each consists of a number of subgrammars, e.g. to form full clauses (with finite verb), to form full sentences, to form 'small clauses' (with no finite verb). In the case of a construction containing *intelligent* it should be possible to derive each of the sentences in (20).

(20a) He seems intelligent.

(20b) He seems to be intelligent.

(20c) It seems that he is intelligent.

The derivations begin with an adjectival proposition (21a), which is turned by one subgrammar into a clause (21b) and then into either an infinitival clause (21c) or into a 'full' sentence (21d). In another subgrammar it is transformed into a 'closed' adjectival formation (21a).

(21a) he intelligent

(21b) he be intelligent

(21c) he to be intelligent

(21d) (that) he is/was intelligent

Tense and aspect are applicable in the clause and sentence subgrammars but not in the adjectival subgrammar. Each of these formations is inserted into a verbal phrase construction (*seem x2*), producing (22a-c), from which appropriate transformations moving *he* to subject position or inserting *it* yield the sentences in (20).

(22a) seem he intelligent

(22b) seem he to be intelligent

(22c) seem that he is/was intelligent

In Dutch there are parallel subgrammars for the corresponding adjectival formation (23a), and for producing an infinitival clause (23b) or a sentence (23c).

(23a) *hij intelligent*

(23b) *hij intelligent te zijn*

(23c) *dat hij intelligent is/was*

Either can be arguments of *schijnen* which thus permits the eventual production of (24b) and (24c) corresponding directly to (20b) and (20c). There is no Dutch equivalent for (20a) because *schijnen* cannot have a 'closed' adjectival complement (24a).

(24a) **Hij schijnt intelligent.*

(24b) *Hij schijnt intelligent te zijn.*

(24c) *Het schijnt dat hij intelligent is.*

It should now be clear how Rosetta deals with constructional mismatches of the kind exemplified in (25) and (26).

(25a) *Hij schaamt zich ervoor.*

(25b) He is ashamed of it.

(26a) *Hij is mij 3 gulden schuldig.*

(26b) He owes me 3 guilders.

In Dutch the two-place function *schaam* generates reflexive verb forms in clauses and sentences, where in English the corresponding two-place function *ashamed* generates adjectival constructions; in Dutch the operation of a subgrammar producing an adjectival output is blocked, and in English the verbal subgrammar is unproductive. Likewise for the three-place function *owe*: in Dutch an adjectival subgrammar is successful, in English a verbal subgrammar.

The division of M-grammars into subgrammars is motivated by the need (particularly in a project involving a team of researchers) for transparent modularity. The modular approach implies the explicit definition for each subgrammar of what is used from other subgrammars (import) and what is to be used by other subgrammars (export). In addition, the rules of any one subgrammar are local to that subgrammar. In Rosetta, the subdivision was inspired by the notion of 'projection' from the $\overline{X}$ formalism (section 2.9.4). As we have seen, for every grammatical category there are tensed and tenseless propositional constructions — these are projections of the categories; hence, in the relevant subgrammars, the imports are S-trees with these categories as heads and the exports are S-trees with their projections.

There is a further argument for modularity in Rosetta. In the M-grammar formalism the explicit ordering of rules is not possible. Rules may be ordered implicitly by splitting a single syntactic category (e.g. NP) into several arbitrary categories (e.g. NP1, NP2, NP3, etc.) and by giving the rules applicability conditions which ensure the desired ordering (e.g. a rule transforming NP1 into NP2 must precede one transforming NP2 into NP3.) It is argued that the subgrammar approach to modularity provides a 'natural' way of expressing the application order of rules.

16.7 Rule classes

In addition to the division of M-grammars into subgrammars, there is also the distinction already mentioned between 'meaningful' syntactic rules and transformation rules. The latter are those rules which are specific to a particular language and serve only a syntactic function. In the earlier Rosetta2 the strict application of the isomorphism principle entailed the inclusion of such rules in grammars of other languages where they served no function. But in Rosetta3 only the meaningful 'translationally relevant' syntactic rules are subject to the isomorphy condition. These meaningful rules are formed into classes of rules handling types of linguistic phenomena, e.g. valency relations, scope, time, voice, negation. Further structure is introduced by restricting translation relations to correspondences between rule classes; only those rules of different languages which belong to the same meaningful rule class may correspond to each other, and hence rules not belonging to the same meaningful rule class cannot be translations of each other.

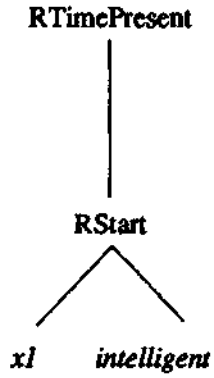
This classification into rule classes cuts across the division into subgrammars. In the *graag/like* example (16), the different structures are handled by different types of subgrammars (adverb and verb), but they involve the same meaningful rule class. It is proposed in Rosetta, therefore, to divide the rules of subgrammars

into 'rule subclasses' and to define the application sequences of rules in terms of these rule subclasses. There are many similarities among rule classes in different subgrammars. For example, all subgrammars contain rule classes for valency and negation. Furthermore, as example (20) above showed, the subgrammars for the same head category are broadly equivalent in meaning. It is suggested that subgrammars may also be isomorphic within a language, in the sense that they correspond with respect to their meaningful rules. As a consequence, the derivation trees for (27a) and (27b) include the same sequence of meaningful rules, while being produced by different subgrammars; in the former (28a) by a subgrammar for the projection of adjective to 'adjectival phrase' (ADJPPROP), and in the latter (28b) by a subgrammar for the projection of adjective to clause.

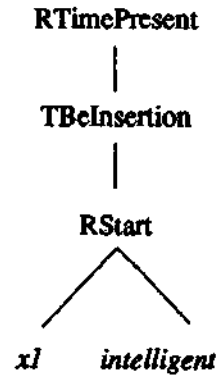
(27a) the intelligent girl

(27b) The girl that is intelligent.

(28a) ADJPROP



(28b) CLAUSE



It is assumed that the time reference is needed in both, although not realised in (27a), not just for isomorphy reasons but also with a view to model-theoretic semantic interpretation (cf. section 16.2 above).

The notion of attuning subgrammars to each other applies also to subgrammars with different head categories. We have seen it already in example (16). However, the situation is rather more complex than in (27)–(28). The two subgrammars for clause projections of ADV and VERB cannot be completely isomorphic, but parts of them can be: those concerned with verb valency (*xl swim/zwem*), with tense, and with the insertion of subject (RSubst1 and RSubst2). There can thus be partial isomorphy of subgrammars of different languages, and perhaps (it is claimed) complete isomorphy between sets of subgrammars.

16.8 Lexical transfer

The isomorphic requirement would appear to present intractable problems for translational equivalences where there can be no correspondence of component

lexical items. The most obvious examples are idiomatic expressions. Consider the equivalent pair of idioms (29) in English and Dutch.

(29a) spill the beans

(29b) *zijn mond voorbij praten*
lit. 'one's mouth past talk' ('to talk past one's mouth')

The problem is that the non-literal meanings of these expressions cannot be derived compositionally from their elements, and yet the expressions have to be translated as wholes. However, it will be recalled that basic expressions in Rosetta do not necessarily have to be single lexical items. Thus, fixed idioms such as *red herring*, *by and large*, *kant en klaar* ('ready-made') can be treated as units. Most idioms, however, cannot be treated as strings: they are structures which vary according to context (30).

(30a) Peter broke Mary's heart.

(30b) Peter was breaking his sister's heart.

(30c) Mary's heart was broken by Peter.

It is argued that idioms should be represented in Rosetta as normal syntactic structures, but at the same time as single 'basic expressions'. As a consequence, Rosetta dictionaries will contain both basic expressions with no internal structures (i.e. a primitive meaning) and also complex basic expressions that do have internal structures, defined perhaps by a subset of the grammar. In those cases where a particular expression has a literal as well as an idiomatic meaning ((29a) and *kick the bucket*), then the grammar will derive meanings compositionally in addition.

This approach is believed to be capable of dealing with problems of lexical transfer such as equivalences between simple verbs such as Spanish *madrugar* and the phrases *get up early* and (Dutch) *vroeg opstaan*. The apparent implication is that the Dutch and English phrases will be assigned a 'basic meaning' corresponding to that of the Spanish verb, i.e. they are treated as 'translation idioms'. The approach is clearly motivated by the desirability of maintaining isomorphism. In a similar vein, the translational equivalences of (31) and similar constructions are tentatively handled by the positing of an 'abstract preposition' in Spanish with the same meaning as *across/over*.

(31a) He ran across the square.

(31b) *Hij rende het plein over.*

(31c) *Cruzó la plaza corriendo.*

In the generation of Spanish there would be transformation rules (i.e. specific to Spanish) which change structures containing this preposition into structures with *cruzar* and a gerund form of the movement verb.

16.9 Comments and conclusions

In the last two sections we have seen some of the 'contortions' into which adherence to the isomorphism principle has led the Rosetta researchers. It has meant the acceptance of partial and overlapping subgrammars to deal with similar but not identical structures within and between languages, and it has meant the creation of elements or structures which cannot be motivated monolingually. What

has happened is that as the Rosetta grammars have been extended to deal with larger ranges of language phenomena, and the strict isomorphy of the earlier Rosetta2 design has given way to a looser conception. There are now, as we have seen, two types of rules in M-grammars, meaningful rules and meaningless rules. The former are interlingual, in the sense that they correspond to 'rule meanings' in semantic derivation trees; the latter are language-specific, they correspond with nothing in intermediate representations and they are not isomorphic with rules in grammars of other languages.

Although compositionality and isomorphism are clearly the bedrock of Rosetta, and mutually support each other, they seem sometimes to be in conflict. Consider further the *madrugar/get up early/vroeg opstaan* examples: isomorphism demands that the Dutch and English phrases receive a 'basic meaning' corresponding to that of Spanish *madrugar*; compositionality requires the meanings of the Dutch and English phrases to be derived from the meanings of constituent 'basic expressions' (*get up* and *early*, and *opstaan* and *vroeg*) together with the meaning of the structural relation involved. It would appear to an outside observer that the only way that the Dutch/English semantic derivation trees can be made isomorphic with the Spanish semantic derivation tree is by the elimination (in some manner) of the basic meanings for *get up/opstaan* and for *early/vroeg* from the trees and retaining only the composite meaning for the phrases as wholes. But this option would not be valid for other *get up* constructions (e.g. *get up at noon*) where there must be correspondence between the basic meanings of *get up* and the Spanish equivalent *levantarse*.

The Rosetta interlingua representation is defined by the isomorphic grammars of the languages of the system. Interlingual elements are explicitly denied universal status. It is easily conceivable that the semantic derivation trees for isomorphic grammars of, say, Japanese and Chinese would differ substantially from the semantic derivation trees for the isomorphic grammars of Dutch and English. The apparatus of Montague semantics appears to have no relevance; Rosetta makes no use of any 'model-theoretic' logical interpretations, which might provide independent justification for semantic derivation trees. Indeed, in most respects the Rosetta design is like any other linguistic-based MT system: morphological analysis, surface syntactic structures, semantic analysis in terms of deep structure relations (valency, tense, voice, predicate–argument structures, etc.). The only essential difference is that whereas other systems equate dissimilar items and structures (by lexical and structural transfer), Rosetta equates the derivational histories of items and structures.

As an interlingua system, Rosetta departs from the usual assumption that source language analysis and target language generation should be completely independent (section 6.7) The isomorphism principle requires that the grammars and the procedures are explicitly oriented towards translation into particular languages. In this respect, Rosetta could quite legitimately be considered a type of direct MT system. The treatment of category mismatches and the postulation of translation idioms would lend support to this characterization.

It may be noted that the definition of isomorphism (section 16.3) appears to leave open the possibility of more than one 'basic meaning' for an expression or

more than one 'meaning rule' for a syntactic operation. This would allow English *wall* to have two meanings if the grammar is to be isomorphic with, say, a German grammar. But if grammars are to be interchangeable in a multilingual system, then the meanings would have to be distinguished in both analysis and generation even when translating from or into a language where this distinction is not made. It is the familiar problem for all interlingua-based systems (section 6.7.2) The transfer-based answer is, of course, ruled out by the isomorphism principle.

Reversibility itself raises difficult problems of control. Some control of rule application is introduced by subgrammar modularization (section 16.6), but there is still the question whether all rules should be optional. The M-grammars are free production systems but computational efficiency favours the inclusion of obligatory rules. Unfortunately a rule which should be obligatory during generation may have to be optional during analysis, and it is unclear how this can be handled in reversible grammars. However, it is not a problem unique to Rosetta.

There are particular problems in trying to evaluate the Rosetta system. On the conceptual level it is difficult to grasp the potential advantages or disadvantages of the approach because of the interaction of innovative theoretical principles, most of which have not been adopted previously in MT research (only a weak form of compositionality in Eurotra, see Chapter 14). So far only small demonstration grammars have been developed and the success of their computational implementation is not known. Indeed, there have been few internal reports from Rosetta on computational aspects, and these are concerned mainly with the formalization of reversible parsers and generators — it is even unclear which programming language is to be employed, although the obvious candidate would seem to be Prolog.

In a practical implementation Rosetta is intended to operate interactively with users. It would seem that at present interaction is envisaged only during analysis, as an aid to disambiguation. There would appear to be no intention of restricting generation to a single output, e.g. the 'best' translation according to some internally specified criterion. In the present experimental phase it seems that, for example, both the legitimate sentences in (24) would be produced for any one of the input sentences in (20), and vice versa. What has been suggested for a later phase of development is the addition of a generation module which would convert 'unnatural' output into more natural 'paraphrases' — although the consequence would be asymmetry of analysis and generation processes and a loss of isomorphism.

Despite any reservations there may be about the practicality of the Rosetta approach, and the project is at too early a stage for realistic evaluation, the major contribution of Rosetta has been and will remain the solidly based exploration of a highly principled approach to translation and the consequential expansion and enrichment of MT theory. It has stimulated many MT researchers to consider more thoroughly the foundations of their own approaches to MT, which must surely be to the benefit of future research.

16.10 Sources and further reading

The theoretical content of this chapter is based on the descriptions by Appelo and Landsbergen (1986), Landsbergen (1987a,b), all of which provide the substantial formalism which underpins the foundations of Rosetta and which for space reasons has been omitted in this account. For somewhat simplified descriptions of the linguistic basis readers may consult Leermakers and Rous (1986), Sanders (1988) and Landsbergen *et al.* (1989). The description of Rosetta subgrammars is derived from Appelo *et al.*(1987) and Odijk (1989); and the treatment of idioms is covered by Schenk (1989) and by Landsbergen *et al.* (1989)