

WPFG

working papers in functional grammar

WFG no. 18
November 1986

Two papers on the computational application of FG
Simon Dik
University of Amsterdam

Two papers on the computational application of FG

Simon Dik
University of Amsterdam

Preface

This Working Paper combines two studies on the potential computer applications of Functional Grammar.

The first paper was read at the ICAME conference held in Amsterdam, June 1986; the second paper was a contribution to the 3d Workshop on Natural Language Generation, held in Nijmegen, August 1986.

I am grateful to Nettie Theyse for producing the second paper in its present form.

Simon Dik.

Simon C. Dik
Institute for General Linguistics
University of Amsterdam.

0. Introduction

In the spirit of the workshop for which it was prepared this paper gives a brief and sketchy impression of some aspects of the theory of Functional Grammar (FG) which might be relevant to the computational handling of natural language data. Some computationally oriented work using FG has already been done (e.g. Kwee 1979, Weigand 1984, 1986), and some other projects are on their way. But on the whole, computational application of FG is still very much in an incipient stage. If I may seem to overstate the computational potential of FG, this should be taken as intended in a programmatic sense.

1. Sketch of FG

Functional Grammar, as developed in Dik (1978a) and later publications, is a general theory of the organization of natural languages, based on a functional perspective on the nature of language. FG strives for optimization with respect to the following standards of adequacy:

- (i) Typological adequacy: the theory should be formulated in terms of rules and principles which can be applied to any type of natural language.
- (ii) Pragmatic adequacy: what the theory says about a language should be such as to help us understand how linguistic expressions can be effectively used in communicative interaction.
- (iii) Psychological adequacy: what the theory says about a language should be compatible with (what is known about) the psychological mechanisms involved in natural language processing.

The standards (ii) and (iii) might also be captured by saying that FG should be a good candidate for providing the grammatical module for a naturalistic model of a natural language user (NLU). Requirement (i) boils down to the idea that such a model is more interesting and tells us more about the basic principles underlying the faculty of language to the extent that it is able to cope with languages of a variety of types.

2.

The organization of FG can be represented in rough outline as in Figure 1:

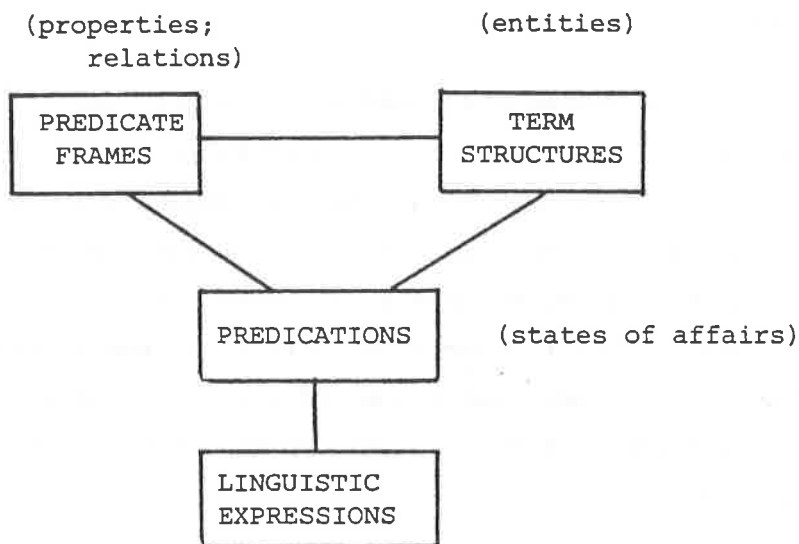


Figure 1. Global structure of a Functional Grammar.

Linguistic expressions are analyzed in terms of more abstract underlying predications, which are taken to designate various types of States of Affairs (Actions, Processes, States, Positions - in general, things that can be the case in some real or potential world); these predications consist of predicates (designating properties or relations) and terms (designating entities). All predicates are contained in predicate frames, which define their essential semantic and combinatorial properties. The predication is the central construct in terms of which the formal and semantic properties of linguistic expressions are defined, and in terms of which the generalization over different language types is achieved. In actual fact, the organization of FG is rather more complex, as indicated in Figure 2.

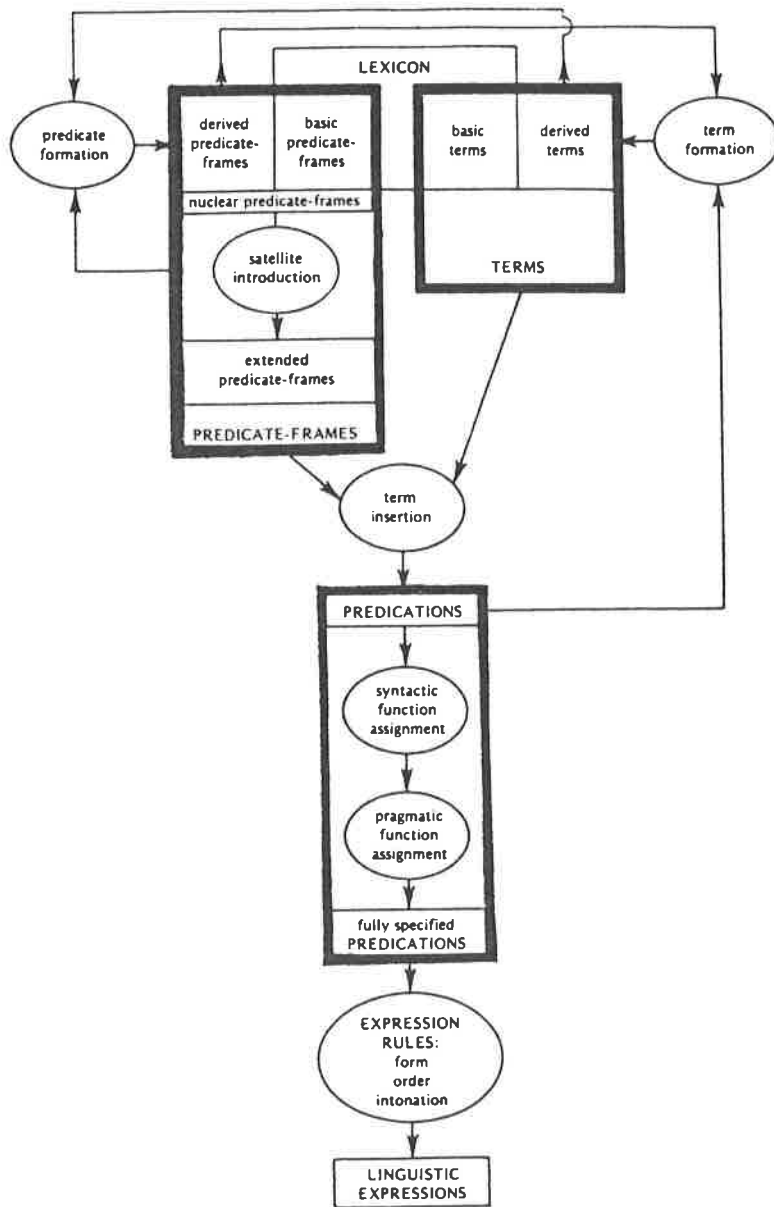


Figure 2. More detailed lay-out of a Functional Grammar

We see here that each of the basic modules of FG--predicate frames, term structures, and predications -has a certain amount of internal complexity. For example, predicate frames may be either basic (contained in the lexicon) or derived (formed by productive rule); and in each case, a 'nuclear' predicate frame may be extended by positions for 'satellites' (= circumstantial modifiers) of various types. Terms, again, may be basic or derived. For example, a personal pronoun such as he/him is basic, but most other terms are construed by productive rules, and may take any measure of complexity. Thus, in one case we may want to refer to some entity by means of he, in some other case we might use a complex term such as:

- (1) that boy with reddish hair standing in the corner with a
 glass of wine in his hand

Predications, again, may be specified with various kinds of operators and functions specifying their different potential semantic and grammatical properties.

2. The predication: an example.

Consider linguistic expression (2B) as produced in the context of (2A):

- (2) A: For who on earth is Patrick cooking supper ?
 B: Patrick is cooking supper for CYNTHIA.

Note that the intonation structure and the communicative effect of (2B) is partially dependent on the context in which it is produced. Therefore, the fully specified underlying predication of a linguistic expression can only be determined in context: FG analyzes linguistic expressions in context. Ideally, FG should be developed into a model which can handle longer stretches of text or discourse.

We can now give the FG underlying predication for (2B):

- (3) DECL
 {PresProg cook_V
 (d1x_i: Patrick_{pN(x_i)}) AgSubjTop
 (d1x_j: supper_{N(x_j)}) GoObjTop
 (d1x_k: Cynthia_{pN(x_k)}) BenFoc }

This underlying predication says that the whole expression is a declarative (DECL) sentence, built around the main verbal (V) predicate cook, which takes two arguments in the semantic functions Agent and Goal (= Patient), and which in this case is extended by a satellite in the semantic function Beneficiary. The main predicate carries the predicate operators Present and Progressive for Tense and Aspect, respectively. The three variables x_i , x_j , and x_k represent the three entities that take the three semantic roles opened up by this extended predicate frame. Term operators specify these entities as definite (d) and singular (1). The first term structure in (3) can thus be paraphrased as:

- (4) definite singular entity x_i such that x_i has the property of being called 'Patrick'.

Note that Patrick, supper, and Cynthia are themselves analyzed as predicates of the categories Noun (N) and Proper Noun (pN), which have their own predicate frame in the lexicon.

By the side of their semantic function, terms may carry the syntactic functions Subject and Object, and the pragmatic functions Topic and Focus. Subj/Obj are taken to specify the 'perspective' from which the state of affairs designated by the predication is presented; formally, the assignment of Subj and Obj may lead to active/passive realization of the predication. Top/Foc are taken to specify the informational status of the constituents to which they attach. Thus, Cynthia is specified as Focus in the context of (2), since this term answers the question word in (2A), and thus presents the most essential or salient information in the answer. Formally, this leads to greater intonational prominence of this constituent.

Note that the predication is not taken to be linearly ordered: any permutation of the components of (3), even in two- or three-dimensional format, would be considered as representing the same predication. Linear order is created through the expression rules which map underlying predications onto the actual strings of words that express them.

3. Predicate frames and meaning definitions

Each predication is built from a predicate frame through term insertion and operator/function specification. The predicate frame for cook, as a basic predicate, will be part of the lexicon. Its entry could be represented as follows:

$$(5) \quad \text{cook}_V (x_1 : \langle \text{human} \rangle (x_1))_{\text{Ag}} (x_2 : \langle \text{food} \rangle (x_2))_{\text{Go}}$$

$$= \text{df}$$

$$\text{prepare}_V (x_1)_{\text{Ag}} (x_2)_{\text{Go}} (x_3 : [\text{heat}_V (x_1)_{\text{Ag}} (x_2)_{\text{Go}}] (x_3))_{\text{Manner}}$$

This entry specifies the following information for cook: it is a verbal (V) predicate taking two arguments in the functions of Agent and Goal. The Agent position is selectionally restricted to terms indicating human beings, the Goal position is reserved for terms indicating some kind of food. The meaning of cook is defined as 'to prepare by heating'. This is again captured in a predicate frame which says that when x_1 cooks x_2 this is equivalent to saying that x_1 prepares x_2 in a Manner x_3 which is defined by x_1 heating x_2 .

The definiens of a meaning definition such as (5) obviously contains other predicates of English, just as an ordinary dictionary does. These predicates, however, have their own entry in the lexicon, such as:

(6) $\text{prepare}_V (x_1 : \langle \text{animate} \rangle (x_1)) \text{Ag} (x_2) \text{Go}$

=df

$\text{make}_V (x_1) \text{Ag} (x_3 : [\text{ready}_A (x_2) \emptyset] (x_3)) \text{Go}$

(7) $\text{heat}_V (x_1) \text{Ag/Force} (x_2) \text{Go}$

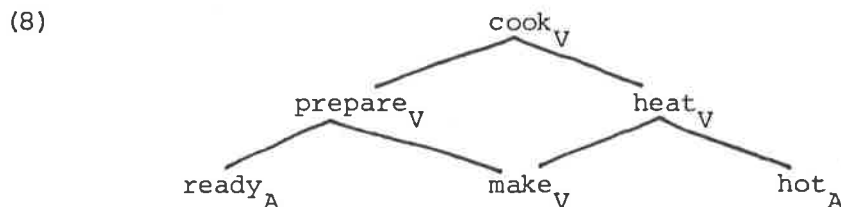
=df

$\text{make}_V (x_1) \text{Ag/Fo} (x_3 : [\text{hot}_A (x_2) \emptyset] (x_3)) \text{Go}$

In (6) it is said that when x_1 prepares x_2 this is equivalent to saying that x_1 'makes' (= causes to come about) a state x_3 defined by x_2 being ready. Note that heat in (7) has precisely the same pattern.

Through this method of 'stepwise lexical definition' (cf. Dik 1978b) we analyze a given predicate in terms of combinations of more simple predicates, and these again into even more simple predicates, until finally we arrive at predicates which cannot be defined any further without resorting to circular definitions.

The meaning definitions together implicitly define a network of semantic relations in the lexicon. For example, the meaning definitions (5)-(7) define the following network:



If it is true that this type of stepwise definition is not only a matter of practical dictionary writing, but also the most adequate way of theoretical accounting for lexical meaning in FG, then obviously existing dictionaries (both mono- and bilingual) contain an enormous wealth of semantic information, which should be more systematically exploited in semantic analysis than is now commonly the case.

As dictionaries are increasingly available in a form that can be processed by computer, this opens up an enormous fund of possible research into

8.

semantic relations. This idea underlies research on 'links' in the lexicon, designed by Willem Meijs and briefly described in Meijs (1986).

In fact, a dictionary can be regarded as a huge corpus, different from corpora consisting of running text, but no less rich in information concerning the language involved. Obviously, this is not to say that the definitions found in current dictionaries are necessarily correct. On the contrary, most lexical definitions contain either too much or too little information, or even both at the same time, in respect to the semantic information which is necessary and sufficient to account for the different uses of the word defined. All the same, current dictionaries offer the best approximation on a grand scale of that which a person has to know if he is to 'understand' the words of a language.

Note that the type of meaning definition advocated here provides us with a powerful instrument for automatically generating different paraphrases of given linguistic expressions. For example, the definitions (5)-(7) define the following expressions as (more or less) semantically equivalent:

- (9) a. Patrick is cooking supper
b. Patrick is preparing supper by heating it
c. Patrick is preparing supper by making it hot
d. Patrick is making supper ready by heating it
e. Patrick is making supper ready by making it hot

Furthermore, these lexical definitions allow us to draw various inferences from given information. For example, if it is the case that Patrick has fully achieved the action of cooking a steak, then we may conclude that the steak must be ready and hot. Such inferences play an important role in natural language understanding.

Apart from this role in lexical definition, the predicate frame is an important tool in the production and the analysis of linguistic expressions. As for production, it is clear that the predicate frame, as stored in the lexicon, provides us with a ground plan for building predications: the skeleton structure of the predication is already contained in the predicate frame.

As for analysis, a powerful parser could be built on the principle that first, the (main) predicate of a linguistic expression is identified, then the predicate frame of that predicate is retrieved, and finally the rich information contained in that predicate frame is used to identify and interpret the other constituents contained in the linguistic expression.

For example, suppose we wish to analyze and interpret a sentence such as:

(10) Patrick is cooking fufu.

and suppose, further, that we do not know what fufu is. Even then, retrieving the predicate frame for cook as given in (5) will help us greatly in arriving at an interpretation of (10). That predicate frame tells us that with cook we should expect a human Agent and some kind of food in the function Goal. In the lexicon Patrick will be defined as 'name of male person', and person as 'human being'. Therefore, Patrick must represent the human Agent. The grammatical information (initial position in an active construction) is compatible with that conclusion. This being so, fufu is most probably the Goal term and must therefore indicate some kind of food. Most probably, since in theory (10) might also represent a case in which cook is used in an absolute sense, and fufu indicates some kind of adverbial modifier. And obviously, even more far-fetched interpretations are conceivable. It remains true, however, that the richly structured information stored in the lexicon is of great help in analyzing and interpreting linguistic expressions. Automatic parsing should therefore preferably take the form of an interactive to and fro between linguistic expression and lexicon.

4. Widening the scope

I believe it has been established that the notions of predicate frame and predication, as well as the general lay-out of FG, carry us a long way towards reaching the goal of a typologically adequate model of grammar. If this is true, then one might expect that something like a functional grammar also plays an important role in the cognitive organization that allows human beings to communicate through natural languages. As we saw in the introduction, a model that is to be pragmatically and psychologically adequate should be a good candidate for providing the grammatical module in a model of the natural language user (NLU). We could thus widen the scope of grammatical research by giving pride of place to the question: how could we build a naturalistic model of NLU? And, by way of thought experiment, we could think of answers to this question in terms of computational implementation. Let us thus think about devising a computational NLU (CNLU) which is to be able to interpret and produce NL expressions in a natural and communicatively effective way.

Some minimal requirements on CNLU can be represented as in Figure 3:

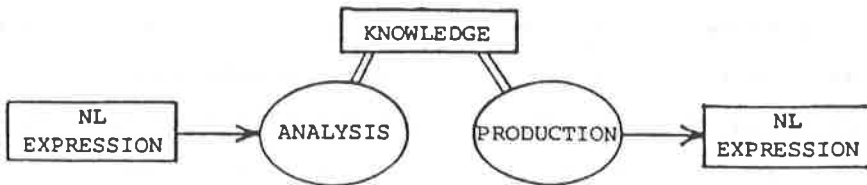


Figure 3. Minimal components of a Natural Language User.

In other words, CNLU is to be able to analyze NL expressions, relate these to the knowledge stored in its system, and produce an NL expression as a response which is contextually relevant as monitored through this knowledge.

Obviously then, knowledge plays a central role in relating linguistic input to appropriate linguistic output. What form does knowledge take? How might one imagine knowledge to be represented in CNLU? In Dik (1986) I have given a linguistically motivated answer to this question in the form of the following hypothesis:

- (11) Knowledge representation.
- (i) knowledge is either perceptual or conceptual;
 - (ii) perceptual knowledge takes the form of perceptual images;
 - (iii) conceptual knowledge takes the form of predications.

According to (11) (iii), then, our non-perceptual knowledge is stored in the same format as the underlying predications in terms of which NL expressions can be analyzed and produced.

Let us consider the advantages of this hypothesis. Consider the following exchange:

11.

- (12) A: How many employees does Johnson have ?
B: (i) Johnson has 35 employees.
(ii) 35.

On our hypothesis, the appropriateness of B's answers would be established in the following way: B first constructs the predication underlying A's question:

- (13) $INT \left\{ \begin{array}{l} \text{Pres have}_V (d1x_i : \text{Johnson}_{pN}(x_i)) \emptyset_{\text{SubjTop}} \\ (Qnx_j : \text{employee}_N(x_j)) \text{GoObjFoc} \end{array} \right\}$

where INT stands for the interrogative illocution, and Q is a question operator applying to the number slot n of the second term.

Suppose now that B has the following knowledge stored in his memory:

12.

- (14) Pres have_V (d1x_i : Johnson_{PN}(x_i)) ∅
(35x_j : employee_N(x_j)) Go

By means of simple matching operations, this knowledge could then be identified as containing the information requested by A, and B could thus form the underlying structures for the answers (12)B(i) and (ii), as follows:

- (15) (i) DECL { Pres have_V (d1x_i : Johnson_{PN}(x_i)) ∅SubjTop
(35x_j : employee_N(x_j)) GoObjFoc }
- (ii) DECL { 35_{Foc} }

where (15) (ii) would be determined by a pragmatic rule which says that it is sufficient for an answer to a Q-interrogative to contain just that information which 'fills in' the slot on which Q operates in the question.

The general point here is that if NL understanding requires us to relate linguistic expressions to knowledge, then it would be very useful indeed if that knowledge were stored in a form similar or identical to the structures underlying the NL expressions. And further: there is, as far as I can see, no principled reason why this should not be the case. In other words, the predication would seem to be an excellent carrier of knowledge.

Obviously, matters are much more complex than the simple example given above would suggest. For one thing, many different types of knowledge are involved in NL communication. In Dik (1986) I have suggested a typology in which ten different types of knowledge are distinguished. I will not go into this matter of knowledge typology here.

For another thing, if we accept hypothesis (11), then we must also accept that the form in which the 'same' knowledge is stored is not necessarily uniquely determined. In other words, you and I may know the same thing in different forms. For example:

- (16) a. Your knowledge: Peter Smith has a daughter called Mary.
b. My knowledge : Mary Smith's father is called Peter.

Our hypothesis requires us to accept that (16a) and (16b) are different pieces of knowledge, although they boil down to the same factual relation between Peter Smith and Mary Smith. This means that either our hypothesis

is wrong (in which case we must try to find some language-independent way of representing the 'common' knowledge contained in (16a-b)), or we need a device which allows us, given some piece of knowledge, to derive other, equivalent pieces of knowledge. Convinced of the attractiveness of the linguistic approach to knowledge embodied in hypothesis (11) I shall follow the latter course.

For another example of the non-uniqueness of knowledge representation, compare the following:

- (17) a. Johnson has 35 employees.
- b. Johnson employs 35 people.
- c. There are 35 people working for Johnson.
- d. Johnson gives work to 35 employees.
- e. Employer: Johnson; employees: 35.

In some sense all these expressions contain the 'same' knowledge. Our hypothesis requires us to accept, however, that one person may know the relevant fact in one form where another person knows the same fact in another form; and we can accept this, provided that we have a mechanism for mutually deriving equivalent pieces of knowledge from each other.

5. Functional Logic

We saw at several points above that CNLU needs a mechanism through which valid conclusions can be drawn from given premisses. If linguistic expressions are analyzed in terms of underlying predications of the FG format, and if knowledge is represented in that same format, then this mechanism must be such as to accept predications as input and deliver predications as output. This mechanism may be called a Functional Logic. First suggestions for such a Functional Logic have been made in Brown (1985).

Let us consider one important property of Functional Logic here: if Functional Logic is a system which takes FG predications as input and delivers FG predications as output, then it must be the case that the 'forms' in ^{terms of} which patterns of valid reasoning are defined are identical to the 'forms' in terms of which linguistic expressions are grammatically analyzed. In other words, in the approach sketched here logical form must be identified with grammatical form, where

the grammatical form of a linguistic expression is defined as the predication underlying that expression.

The assumption of the identity of grammatical and logical form leads to an essential departure from Standard Logic, which since Frege and Russell has been based on the Misleading Form Thesis, which says that in many crucial cases the logical form of a linguistic expression is (very) different from its grammatical form.

Let us illustrate this point by a simple example. Consider a syllogism of the following form:

- (18) a. All men are fallible.
 b. Socrates is a man.
 c. ∴ Socrates is fallible.

In Standard Logic, this pattern would be analyzed in terms of the following logical forms:

- (19) a. $\forall x (M(x) \rightarrow F(x))$
 b. $M(s)$
 c. ∴ $F(s)$

and a rule of inference which says that, given the premisses (19a-b), the conclusion (19c) is valid.

Note, however, that the logical forms for (18a) and (18b-c) are quite different. In particular, the logical form for universally quantified expressions such as (18a) departs in essential ways from what, at first sight, would appear to be the grammatical form of such expressions. This means that the mapping of logical form (19a) onto its linguistic expression will necessarily be different from, and much more complicated than, the mapping of logical form (19c) onto its linguistic expression.

In FG, however, the assumption is that the grammatical forms of (18a) and (18c) are essentially similar in that in both cases the main predicate fallible is applied to a term which in the former case is a universally quantified term, in the latter a definite term containing a proper name. If logical form is identical to grammatical form, then a Functional Logic will analyze syllogism (18) according to the following pattern:

- (20) a. $F(\forall x: M(x))$
 b. $M(dix: S(x))$
 c. ∴ $F(dix: S(x))$

Note that although these forms are identical to the presumed grammatical forms of the relevant linguistic expressions, there is no reason to assume

that a valid rule of inference could not be formulated which would define (20) as a valid pattern of reasoning.

6. Translation °)

Finally, a few remarks on the role of the predication in translating from one language to another. Note first that a linguistic expression of L_1 and its closest translation equivalent in L_2 do not necessarily have the same underlying predication. For one thing, the system of grammatical operators in L_2 may differ from that of L_1 . For example, Dutch does not have a Progressive aspect of similar productivity as English and therefore an English Progressive will either be lost in its Dutch translation, or it will lead to a rather different structuring of the predication, as in:

- (21) a. John was reading the paper.
 b. Jan zat de krant te lezen.
 John sat the paper to read = 'John sat reading the paper'

Although (21b) is more or less equivalent to (21a), it requires that 'Jan' be in a sitting position. Compare:

- (22) a. (John and Mary stood on the platform waiting for the train to arrive.) John was reading the paper.
 b. (i) Jan stond de krant te lezen.
 John stood the paper to read
 (ii) Jan las de krant.
 John read the paper

We see that translation (i) is more specific than the English original, whereas translation (ii) is more general, in that the Progressive is left unaccounted for.

Thus, the mapping of grammatical operators from one language to another need not be a matter of simple one-to one equivalence.

In the second place, the lexical predicates of L_1 need not have direct or full correspondents in L_2 . For example, Dutch has no fully equivalent counterpart to the English verb assassinate; usually, this verb will be translated by vermoorden which, however, is equivalent to 'murder' without any suggestion of 'important or prominent victim' or 'for political purposes'. As in the grammatical domain, the closest translation equivalent of a lexical item will often involve an unavoidable loss or gain of information.

This having been said, it is nevertheless true that the predications underlying translation equivalents in two languages are much closer to each other than the linguistic expressions themselves. It is therefore easiest to bridge the gap between two languages at the level of the underlying predication. In other words, we may think of translation in terms of the model sketched in Figure 4:

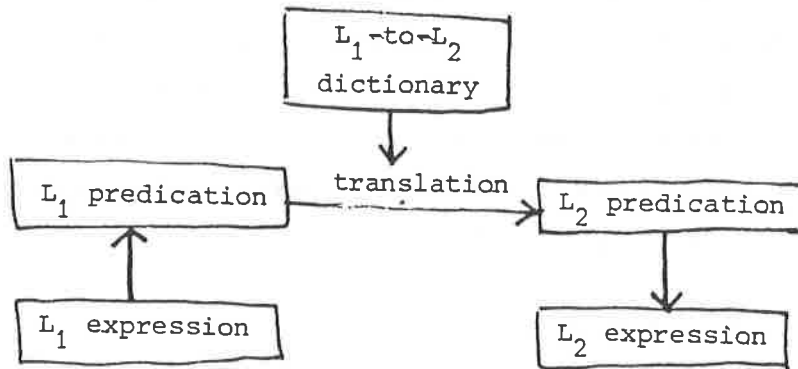


Figure 4. Translating
via the predication level.

In this model, an L_1 expression is first converted into its underlying predication, then this underlying predication is translated into its closest equivalent in L_2 , and finally the L_2 predication is mapped onto its L_2 expression. The conversion from L_1 to L_2 is monitored through an L_1 -to- L_2 dictionary which, for each predicate-frame in L_1 , defines its closest equivalent in L_2 .

The underlying predications are thus seen as the intermediary 'language' through which the gap between L_1 and L_2 can be most efficiently bridged.

For an illustration of this model, compare a Latin sentence and its English translation:

- (23) a. Illos temperaturos ab iniuria non existimabat.
 them refrain-fut-acc from injustice not believed-3sg
 b. He/she did not believe that they were going to refrain from
 injustice.

It is clear that the organization of (23a) is quite different from that of (23b). It would be difficult, then, to map (23a) onto (23b) through any direct word-for-word method. Suppose, however, that we first reconstruct the underlying predication for (23a):

- (24) DECL { Past Neg existimare_V (d1x_i: 3p(x_i))_{Pos}
 (x_j: [Prosp temperare_V (dmx_k: 3p(x_k))_{Ag}
 (x₁: iniuria_N(x₁))_{Source}] (x_j))_{Go} }

Suppose further that we have a Latin-to-English dictionary which gives us the following information:

- (25) a. existimare_V (x₁)_{Pos} (x₂)_{Go} = believe_V (x₁)_{Pos} (x₂)_{Go}

- b. $\text{temperare}_V (x_1)_{Ag} (x_2)_{Source} = \text{refrain}_V (x_1)_{Ag} (x_2)_{Source}$
 c. $\text{iniuria}_N (x_1)_{\emptyset} = \text{injustice}_N (x_1)_{\emptyset}$

This dictionary would allow us to replace the Latin lexical predicates in (24) by their English equivalents by simple substitution, yielding:

- (25) $\text{DECL} \left\{ \begin{array}{l} \text{Past Neg believe}_V (d1x_i : 3p(x_i))_{Pos} \\ (x_j : [\text{Prosp refrain}_V (dmx_k : 3p(x_k))_{Ag} \\ (x_1 : \text{injustice}_N (x_1))_{Source}] (x_j))_{Go} \end{array} \right\}$

This underlying predication can now be mapped onto (23b) and several alternative constructions, according to the expression rules which monitor the mapping of underlying predications onto linguistic expressions in English. One alternative expression form would be the following:

- (26) He did not believe them to be going to refrain from injustice.

which is closer to the original in grammatical organization, but more marked in English than it is in Latin.

Although many complications of this approach to translation are glossed over here, the message, I hope, is clear: the predication is the easiest place to cross the river which divides two languages.

7. Conclusion

I hope I have made it plausible that the notion of underlying predication as developed in FG might be a powerful instrument for the re-unification of grammar, logic, and cognition. Let me recapitulate the potential roles of the predication in a naturalistic NLU model: in such a model, the predication can fulfil the roles of:

- (i) output of a Parser;
- (ii) input of a Generator;
- (iii) instrument for lexical definition;
- (iv) vehicle for conceptual knowledge representation;
- (v) input and output of a Functional Logic inferencing calculus;
- (vi) input and output of a translation module.

So far, only certain aspects of the NLU model sketched in this paper have been sufficiently formalized to allow of computational implementation. I see no principled reason, however, to assume that further formalization of the model should be impossible. This is the basis for my conviction that this FG approach to developing CNLU constitutes a fruitful and promising research programme.

- °) For implementation of a translation procedure via the level of the predication, as sketched in section 6 above, see now Van der Korst (1986), referenced in the next paper.

REFERENCES

BROWN, D. Richard

- 1985 Term operators. In: A.M. Bolkestein et al. eds., *Predicates and terms in Functional Grammar*, 127-145. Dordrecht: Foris.

DIK, Simon C.

- 1978a *Functional Grammar*. Amsterdam: North-Holland (1981³, Dordrecht: Foris).
- 1978b *Stepwise lexical decomposition*. Lisse: Peter de Ridder Press.
- 1986 *Linguistically motivated knowledge representation*. Amsterdam: Working Papers in Functional Grammar 9.

KWEE Tjoe Liong

- 1979 A68-FG(3): Simon Dik's funktionele grammatika geschreven in algol 68, versie no. 3. Amsterdam: Publications of the Institute of General Linguistics 23.

MEIJS, Willem

- 1986 *Links in the lexicon: the dictionary as a corpus*. Bergen University: *Icame News* 10, 26-28.

WEIGAND, Hans

- 1984 *Basic concepts for a FG natural language interpreter*. Paper, Department of Mathematics and Computer Science, Free University Amsterdam.
- 1986 *Natural language semantics in a conceptual language*. Paper, Department of Mathematics and Computer Science, Free University Amsterdam.

GENERATING ANSWERS FROM A LINGUISTICALLY CODED KNOWLEDGE BASE

Simon Dik

1. INTRODUCTION

The computational processing of natural language data is sometimes approached as a purely practical problem, for which 'anything goes' as long as the results are reasonably acceptable. Given sufficient ingenuity of the analyst, this approach may yield short-term successes, which will, however, be of limited theoretical interest. As a theoretical linguist, I am more interested in attempts at finding more principled, linguistically and psychologically motivated solutions to the problems involved. I also believe that such more principled solutions will yield better long-term practical results in any attempt to attack anything more complicated than the simplest natural language processing puzzles.

For this reason I believe it is essential for computer specialists, psychologists, and linguists to cooperate in this area of research. One aim of such cooperation would be to arrive at 'converging models' of natural language processing, models which are linguistically adequate, psychologically realistic, and computationally feasible. What is brought in from each of these disciplines may teach us a lot about the potentialities and constraints to be respected in the other fields. Where linguistic, psychological, and computational requirements converge, we may be certain to have hit upon some essential feature of natural language processing.

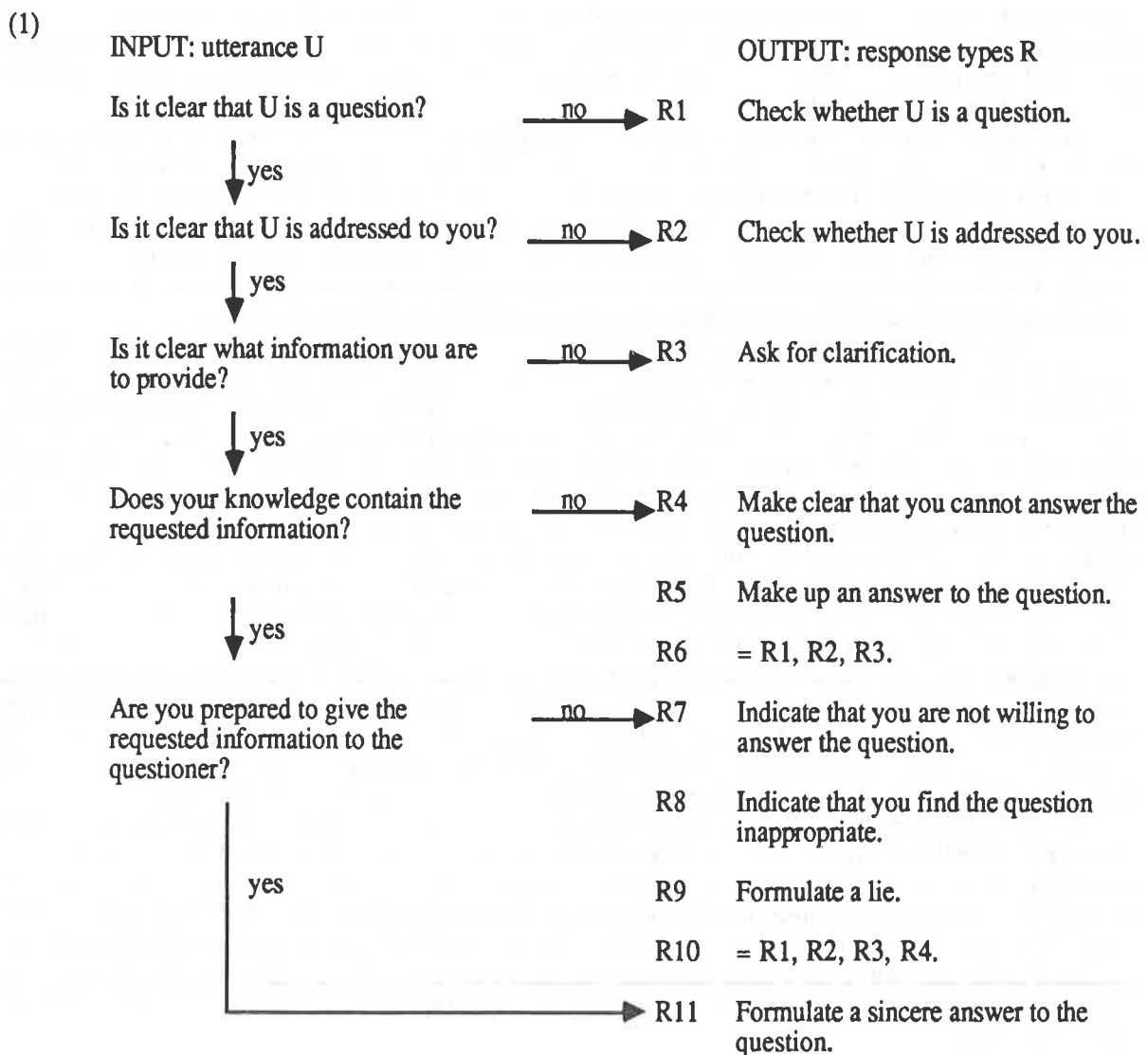
The attempt at arriving at converging models of natural language behavior is especially important for a linguist who, as I do, takes a functional view on the nature of natural language. A view, that is, in which 'form', 'meaning', and 'use' are seen as three components which interactively define the nature of language. This interaction is taken into account in the particular linguistic theory I have been involved in for the last ten years: Functional Grammar (FG) in the sense of Dik (1978) and later publications. FG aspires to develop a model of grammar which is typologically, pragmatically, and psychologically adequate. By typological adequacy we understand that the model should be applicable to natural languages of any type. Pragmatic adequacy means that the model should help us understand how linguistic expressions can be used for communicative purposes. And psychological adequacy requires the model to be compatible with what is known about the ways in which language users process these expressions. An interesting working hypothesis is, I believe, that what is good for typological adequacy is also good for pragmatic and psychological adequacy, and conversely. For example, if we can develop a theory of constituent ordering which can, with suitable parametrization, be applied to any natural language, then this theory may be assumed to also tell us something about how speakers and hearers work in processing and exploiting constituent ordering patterns.

From a functional point of view on language, the most basic question to be asked is: How does the natural language user work? A constructivist version of this question, very useful as a mental exercise for clarifying the many intricate problems involved, is: How could we build a model of a natural language user? And an operational version of the latter question can be formulated as: How could we build a *computational* model of (part of) a natural language user? Looked at in this way, the computational implementation of language processing capacities provides one entry into the underlying theoretical questions that a functionally oriented linguist is interested in.

Against this general background this paper argues that the automatic generation of answers to natural language questions would be simplified if the knowledge required to answer a question would be coded and stored in the same format as the abstract structures underlying the linguistic form of both question and answer. 'Underlying structure' is interpreted in terms of the notion 'underlying predication' of Functional Grammar. If this course is taken, however, it must also be granted that what might seem to be the 'same' knowledge may be coded in a number of different, though cognitively equivalent forms. In order to account for this non-uniqueness of knowledge representation, the question-answering device must be provided with an inference mechanism through which predications (pieces of knowledge) can be inferred from other predications under conditions of logical or probabilistic validity. Some aspects of this inference mechanism are discussed and illustrated.

2. HOW TO ANSWER A QUESTION

Answering a question in an appropriate way is a rather complex business. A question can get many different response types, only some of which can be considered to provide a real 'answer' to the question. Some of the complex decision structure underlying question answering is outlined in the following display:



Among all these response types, only R5, R9, and R11 can be called answers in the sense that they provide the requested type of information; these three answer types differ in their relation to the knowledge/belief structure of the answerer:

(2) Types of answer:

	Answerer replies	Answerer knows/believes
a. Made-up answer	X	\emptyset
b. Insincere answer (lie)	X	Y (\neq X)
c. Sincere answer	X	X

Note that a sincere answer to a question may well be false in a wider, objective sense. Thus, if I have been taught that Acapulco is the capital of Mexico, then (4) is a sincere answer on my part to question (3), even though in fact my statement is false:

(3) What is the capital of Mexico?

(4) Acapulco is the capital of Mexico.

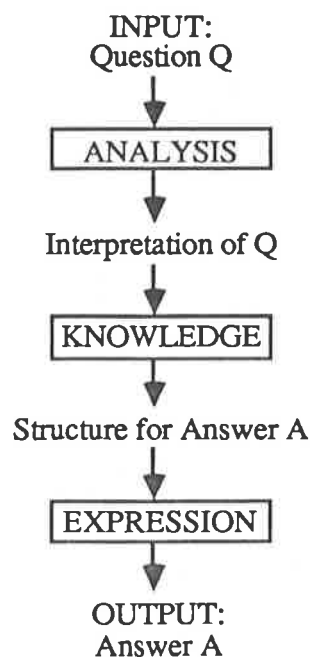
Sincerity is a matter of the relation between an answer and the knowledge possessed by the answerer; truth is a matter of the relation between that knowledge and what is (in some sense) 'really the case'. If we feed a system with incorrect knowledge, then that system's sincere answers may in fact be false. In the same way, an insincere or a made-up answer may in fact be true.

In approaching the problem of how to devise an automated question-answering system we shall not try to tackle the full complexity of (1) in one go. Let us simplify the decision structure somewhat in the following ways:

- (a) We assume that the system understands the question, i.e. understands that it is a question to be answered, and understands what information is to be provided.
- (b) We assume that the system is sincere in that it either provides a sincere answer to our question, or tells us that it cannot do so.
- (c) We assume that the system is prepared to answer our questions if it contains the relevant knowledge.

On these assumptions we can discern the following steps in the question-answering process:

(5)

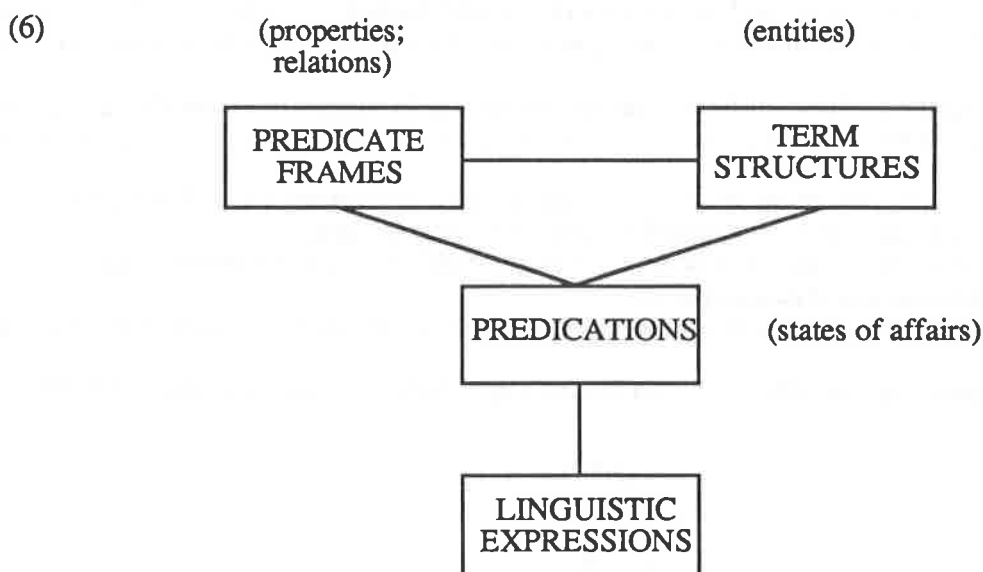


I shall now propose a specific articulation of this minimal and rather trivial system in terms of the principles of the theory of FG.

A model of the natural language user should, to the extent that this is possible at all, be independent of any one particular language. Only in that way do we account for the fact that the human faculty of language is such that any human being can in principle learn to use any natural language. If we conceive a model of the natural language user as consisting of language-independent and language-dependent modules, we will want to maximize the former as long as possible without blocking the operation of the latter. Ideally, and in the long run, we will have to be able to change our model of a speaker of Dutch into a model of a speaker of Japanese simply by substituting the Dutch-particular module by the Japanese-particular one.

One way in which FG tries to achieve such typological adequacy is through the assumption that the linguistic expressions of any one language can be analyzed in terms of more abstract underlying *predications*, structures which contain all the information required both for the semantic interpretation and the formal constitution relevant to the linguistic expression. The idea is that these underlying predications, while not identical across languages, at least have identical structural properties no matter what type of language is involved. The underlying predications of two semantically equivalent sentences in two different languages are taken to be much more alike than the sentences themselves.

The global structure of FG can be represented as follows:



As indicated in (6), the predications underlying linguistic expressions are taken to designate various types of States of Affairs (Actions, Processes, States, Positions); each predication consists of a *predicate* (designating a property or relation) and an appropriate number of *terms* (designating entities). All predicates are contained in *predicate frames*, which define their essential semantic and combinatorial properties. All lexical items of a given language are treated as either basic predicates or basic terms; derived predicates and terms can be formed through rules of predicate and term formation.

In actual fact, obviously, the organization of FG is rather more complex than this sketch of the bare outlines would suggest. There is no room here, however, for going into all the details. Rather, I will give one simple example to illustrate what a predication looks like and how it is built up. Consider the following expressions of English:

- (7) Q. What was the boy doing?
A. The boy was kissing a beautiful girl

The predication underlying (7A) would have the following form:

- (8) DECL (PastProg *kiss* VFoc(d1x₁: *boy* N (x₁) ∅) AgSubjTop
(i1x_j: *girl* N (x_j) ∅: *beautiful* A (x_j)∅) GoObjFoc)

This predication has been constructed from the following basic predicate frames:

- (9) a. *kiss* V (x₁ : < human >(x₁)) Ag (x₂) Go
 b. *boy* N (x₁ : < human > (x₁)) ∅
 c. *girl* N (x₁ : < human > (x₁)) ∅
 d. *beautiful* A (x₁) ∅

Predicate frame (9a) says that *kiss* is a verbal (V) predicate with two argument positions x₁ and x₂, in the semantic functions Agent (Ag) and Goal (Go = Patient), such that the first argument position is reserved for terms indicating human entities through the selection restriction <human>. *Kiss* thus indicates a two-place relation which can be used to construct Action predications. The same goes for the other predicate frames, where N = nominal, A = adjectival, and ∅ = zero semantic function, indicating that the predicates in question can be used to form State predications.

The rules of term formation allow for the construction of terms such as:

- (10) (i1x_j: *girl* N (x_j) ∅: *beautiful* A (x_j) ∅)
 'indefinite (i) single (1) entity x_j such that (:) x_j has the property "girl" such that (:) x_j has the property "beautiful"'

In (8), two such terms have been inserted into the argument slots of the predicate frame of *kiss*. The resulting structure has been further developed by means of the following elements:

- (11) DECL Predication operator indicating that the predication has declarative illocution.
 Past, Prog Predicate operators specifying the tense as Past and the aspect as Progressive.
 Subj, Obj Syntactic functions leading to an active realization of the predication, interpreted as signaling that the Action in question is presented from the perspective of the Agent.
 Top, Foc Pragmatic functions specifying the informational status of the constituents in question in the given context. Topic: the entity about which the predication says something; Focus: the most essential information contained in the predication, given the context.

It is important to note here that the particular string-like notation in which the predication is presented in (8) is not essential to its nature. Any other notation is just as good, as long it preserves the essential components and their hierarchical relations within the predication. For

example, the predication of (8) could just as well be represented in a list format such as:

- (12) III: DECL
 Tense: Past
 Aspect: Progr
 Pred: *kiss*
 PredCat: V
 PragFn: Foc
 Arg1: Def: d
 Num: 1
 Var: x_j
 Restr1: Pred: *boy*
 PredCat: N
 SemFn: Ø
 SemFn: Ag
 SyntFn: Subj
 PragFn: Top
 Arg2: Def: i
 Num: 1
 Var: x_j
 Restr1: Pred: *girl*
 PredCat: N
 SemFn: Ø
 Restr2: Pred: *beautiful*
 PredCat: A
 SemFn: Ø
 SemFn: Go
 SyntFn: Obj
 PragFn: Foc

Similarly, the same predication could be represented in a tree-like format, expressing either the constituency or the dependency relations. The structure of the predication is that network of elements and relations which is preserved through all these notational variants. The relational network of the predication is supposed to be universally applicable to all natural languages. The functions and operators may be different for different languages, but they are considered to be drawn from restricted universal subsystems of the relevant type. For example, the Tense operator Past is drawn from a restricted subsystem of possible Tense operators valid across languages. The actual predicates figuring in predications (such as *kiss*, *boy*, *girl*, *beautiful* in (8) and (12)) are language-particular elements of lexical organization. However, even these items are monitored by language-independent principles defining the possible articulations of the lexicon across languages.

The theory of FG specifies how underlying predications may be mapped onto actual linguistic expressions through expression rules. These rules specify the form and the order of realization of the constituents, given their structural and functional properties in the predication. A reversed application of these expression rules would be used to parse linguistic expressions, i.e. reconstruct the predications underlying these expressions.

3. APPLICATION TO QUESTION-ANSWER SEQUENCES

Returning to schema (5) we shall now assume that the Analysis comprises at least the reconstruction of the predication underlying the question Q; and that what the Expression does is map the predication underlying the answer onto the linguistic expression constituting the answer. Let us apply this to the following sequence:

- (13) Q. Where does Peter Brown live?
A. Peter Brown lives in London.
- (14) Q. INT (Pres *live* \vee (d1x_i : PB (x_i))_{PosSubj} (Qx_j LocFoc)
A. DECL (Pres *live* \vee (d1x_i : PB (x_i))_{PosSubjTop} (d1x_j : London (x_j))_{LocFoc})

Live is analyzed as a predicate designating a State of Affairs of the semantic type Position; its first argument bears the semantic function Positioner (Pos). The predication operator INT indicates that the predication has interrogative illocution. This operator is common to both Yes-No questions and to specific or question-word questions. Question-word questions such as (13Q) are interpreted as 'open' predications in the sense that one (or more) term positions are left unspecified. The unspecified term position is marked by the term operator Q. In (13Q) it is the Location argument position which is left unspecified in the predication. The configuration (Qx) Loc is expressed by *where*. The assumption is that any position marked by Q has Focus function, since the question word marks the informationally most essential component of the predication. The whole structure of the question will be interpreted along the lines of:

- (15) I request you to give me the information which is missing in the following predication: 'Peter Brown lives in ...'

In the structure underlying the answer (14A), the term *London* has been inserted into the position marked by Q in the question. That term again receives Focus function because of its salient informational status in the answer; the term *Peter Brown*, on the other hand, receives Topic function because both question and answer are 'about' the entity referred to by *Peter Brown*.

Different, but equivalent answer forms may be defined through the following two rules:

- (16) (i) the Topic may be expressed in pronominal form.
(ii) the answer may consist of only the Focus constituent.

These rules define the following answers as equivalent to (13A):

- (17) a. He lives in London.
b. In London.

Our assumption is that a person (or system) that is to answer (13Q) works in the following way:

- (18) 1. Reconstruct predication underlying Q.
2. Recognize instruction contained in Q.
3. Check knowledge to see whether instruction can be carried out.
4. If so, construct predication underlying A.
5. Express A.

It is clear that, if this type of procedure is to be automated, we need at least a Parser which can map linguistic expressions onto underlying predications, and a Generator which can map underlying predications onto linguistic expressions. Kwee (1979; 1987) has written a program in Algol68 which can be taken as indicating that a Generator of this type is feasible. Work on the relevant type of Parser is in progress. It will here be assumed that such a Parser is feasible as well.

Step 2 in (18) can be automated by interpreting INT and Q as abbreviations for instructions to the system:

- (19) INT =
 a. Go to knowledge base;
 b. Check whether requested knowledge is available there;
 c. If so, formulate structure for answer containing that knowledge.
 Q = requested knowledge is knowledge to fill this position.

It will be clear that the way in which these instructions are to be carried out depends on (i) what knowledge is contained in the answerer's system, and (ii) how that knowledge is represented. This brings us to the question of knowledge representation.

4. KNOWLEDGE REPRESENTATION

In discussing the problem of knowledge representation we shall start from the very simple example (13), suggest a solution for this example, then generalize that solution into a general hypothesis, and finally discuss the implications of that hypothesis.

If a person is to be able to answer question (13Q), then, apart from a number of general conditions such as knowledge of the language and knowledge of the decision structure given in (1), two more specific conditions must be fulfilled:

- (i) he must know a certain person called Peter Brown, to whom he can relate the term *Peter Brown* in the question;
 (ii) he must have knowledge equivalent to 'Peter Brown lives in L', where L is some specified location; for example, he must have knowledge equivalent to 'Peter Brown lives in London'.

If it is correct, as assumed above, that that knowledge is to be used to construct the underlying predication (14A), starting from the underlying predication (14Q), then it follows rather immediately that the most convenient form of knowledge representation would result from the following assumption:

- (20) Knowledge is represented in the form of predications.

On that assumption, the knowledge required to bridge the gap between question and answer in (13) would take the following form:

- (21) $Pres\ live\ \forall\ (d1_{33} : PB(x_{33}))_{Pos}\ (d1_{x98} : London(x_{98}))_{Loc}$

in which x_{33} is that specific entity known by the subject under the name 'Peter Brown', and x_{98} is that specific entity known by him as 'London'. The whole sequence of structures from question to answer would then be as follows:

- (22) QUESTION Q Where does Peter Brown live?
 PREDICATION-Q INT $(Pres\ live\ \forall\ (d1_{x_i} : PB(x_i))_{PosSubj}\ (Qx_j)_{LocFoc})$
 KNOWLEDGE Pres live $\forall\ (d1_{x_{33}} : PB_{33})_{Pos}\ (d1_{x_{98}} : London(x_{98}))_{Loc}$
 PREDICATION-A DECL $(Pres\ live\ \forall\ (d1_{x_i} : PB(x_i))_{PosSubjTop}\ (d1_{x_j} : London(x_j))_{LocFoc})$
 ANSWER Peter Brown lives in London.

It is then also clear that, in this simple case, the predication underlying the answer can be formed by pressing the specified Location term from the knowledge into the position marked by Q in the predication underlying the question. This can obviously be generalized along the following lines:

- (23) When you are to answer a question of the general form:
 (a) INT $(...(Qx)...)$
 check your knowledge for a predication of the form
 (b) $(...(\alpha)...)$
 which matches the predication of (a) except for containing the specified
 constituent (α) in the place of (Qx) ;
 then construct a predication of the form:
 (c) DECL $(...(\alpha)...)$
 and express this as an answer to the question.

I think it can be taken for granted that, as far as this simple example goes, a system provided with facilities for analyzing and producing predications, and provided with knowledge also stored in the form of predications, could be made to automatically provide sincere answers to questions of this type. In other words, it is worthwhile to investigate the implications of the assumption (20) that knowledge is stored in the form of predications. Let us consider some of these implications.

(A) It is generally acknowledged that knowledge can be divided into perceptual and conceptual knowledge. Perceptual knowledge will consist of percepts: images of things perceived by any of the senses. It is clear that percepts do not take the form of predications. My visual image of my house, for example, cannot be thought of as being stored in the form of predications. We must therefore restrict assumption (20) to the conceptual part of knowledge and assume that knowledge consists of perceptual images, predications, and combinations of these.

(B) Since predications are language-specific in that they contain lexical items of the language concerned, our assumption leads to the further assumption that conceptual knowledge representation is language-specific in the same sense. In other words, our assumption rejects the idea that knowledge should be represented in some language-independent cognitive code or alphabet, distinct from the language we speak. But is this a disadvantage? I do not believe so. Note that if there were a language-independent cognitive code, then any operation involving language and knowledge would require linguistic structures to be translated into cognitive ones, and these back into linguistic ones. Our assumption does not require such translation.

(C) On the other hand, since predications are largely identical across languages as far as their structure is concerned, thanks to the typological adequacy of the notion of 'predication', our assumption does not lead to the untenable position that what a speaker of L_1 knows is completely different from the otherwise equivalent knowledge of a speaker of L_2 . In fact, the predication underlying L_1 are rather easily translatable into the equivalent predication of L_2 on a limited but non-trivial scale, such a translation procedure has been implemented by Van der Korst (1986). Therefore, the knowledge embodied in L_1 predications is just as easily translatable into knowledge embodied in L_2 predications. Our hypothesis requires us to accept, therefore, that a piece-of-knowledge-in- L_1 is different from a piece-of-knowledge-in- L_2 , even if the two pieces of knowledge concern 'the same fact'. On the other hand, we assume that knowledge is represented in structures of such level of abstraction that it is, at that level, easy to bridge the gap between two languages.

5. MORE COMPLICATED CASES

Let us now consider some more complicated cases of question-answer sequences:

- (24) Knowledge: Sally is the daughter of Peter.
 Question: Who is the father of Sally?
 Answer: Peter.

Everybody will accept the answer as correct, given the knowledge available; nobody will accept a response such as 'I don't know' from a person who possesses this knowledge. Nevertheless, the answer cannot be construed on the basis of a simple matching operation as described in (23), since the predication contained in the question does not match the predication representing the knowledge. It is clear that, for this gap to be bridged, some inferencing mechanism is required which defines the following as valid:

- (25) a. Sally is the daughter of Peter.
 b. Therefore, Peter is the father of Sally.

On the basis of the conclusion of (25), the question of (24) can be answered through the matching operation of (23).

An inference mechanism which defines certain inferences as valid and others as non-valid, may be called a logic. In the present case we need a logic which takes predications as input and delivers predications as output. Such a logic we shall call Functional Logic (FL). Note that we can say that a certain inference is valid *by virtue of* a logic. We are thus interested in the inferences which can be drawn from (sets of) predications by virtue of FL.

There are now various ways of widening the scope of our system in such a way that sequences like (24) can be taken into account. I prefer doing this through an extension of the notion of 'conceptual knowledge' in the following way:

- (26) To conceptually know X is either to possess a predication ϕ whose content is X, or a set of predications ψ and an inference mechanism FL such that ϕ is validly inferable from ψ by virtue of FL.

This definition of conceptual knowledge allows me to say that if I know that Sally is the daughter of Peter, then I also know that Peter is the father of Sally, provided I possess the logic which allows me to infer the latter from the former.

To give some more content to the notion Functional Logic, let us add the following. A logic can be defined as consisting of a syntax and a semantics. The syntax specifies which formulae are well-formed 'logical forms'. In the case of FL we assume that the syntax is defined by those rules which specify which FG-predications are well-formed. In other words, the logical form of a linguistic expression is equated to the underlying predication of that expression according to FG. Note that, according to this conception of FL, 'logical form' coincides with 'grammatical form', if the latter is taken in the intended sense. The semantics of a logic may be divided into rules of semantic interpretation and rules of inference. In the case of FL the assumption is that the domain of interpretation consists of 'mental models', consisting of perceptual and conceptual information. The conceptual information is itself again coded in FG predications, i.e., in well-formed logical forms of FL. The inference rules define which predications can be validly inferred from given sets of predications. Some examples of such inference rules will be given below.

A logic can also be characterized in terms of what types of elements it defines the logical properties of. In that sense we speak of 'propositional', 'predicate', 'class', and 'relation logic'. In the case of FL we need at least the following levels of logical analysis:

- (27) Components of Functional Logic:
 a. Illocutionary Logic
 b. Predication Logic
 c. Predicate Logic
 d. Term Logic
 e. Lexical Logic

Let us consider how the desired result could be obtained in the case of example (24). It is clear that inference (25), which is essential to a proper understanding of (24), depends on a

certain relationship between the meanings of the lexical items *father* and *daughter*. We shall say that this inference pertains to Lexical Logic.

In our framework the relevant relationship can be established by associating, with each predicate frame, one or more meaning postulates of the following general form:

$$(28) \quad \alpha(x) \rightarrow \beta(x)$$

saying that when α is predicated of x , then it follows that we can also predicate β of x . It will be clear that if a meaning postulate works both ways, the relation between two predicates (or combinations of predicates) can be written:

$$(29) \quad \alpha(x) \leftrightarrow \beta(x)$$

in which case we can say that α and β are semantically equivalent. In the particular case in which α is a simplex predicate and β is a combination of predicates, we can say that β provides a definition or paraphrase of α .

Let us now consider the predicate frames of *father* and *daughter*:

$$(30) \quad \text{a. } \textit{father} \text{ N } (x_1 : \langle \text{male} \rangle (x_1))_{\emptyset} (x_2 : \langle \text{anim} \rangle (x_2))_{\text{Ref}}$$

$$\leftrightarrow$$

$$\textit{parent} \text{ N } (x_1)_{\emptyset} (x_2)_{\text{Ref}} \text{ and } \textit{male} \text{ A } (x_1)_{\emptyset}$$

$$\text{b. } \textit{daughter} \text{ N } (x_1 : \langle \text{female} \rangle (x_2))_{\emptyset} (x_2 : \langle \text{anim} \rangle (x_2))_{\text{Ref}}$$

$$\leftrightarrow$$

$$\textit{child} \text{ N } (x_1)_{\emptyset} (x_2)_{\text{Ref}} \text{ and } \textit{female} \text{ A } (x_1)_{\emptyset}$$

Both predicates are analyzed here as two-place nominal predicates. These structures can be interpreted as follows: 'saying of a male entity that he is father with reference to an animate entity is equivalent to saying that he is a parent of that entity and that he is male'. Likewise for (23b). As a next step, we must establish the relationship between *parent* and *child*. This can be done through the following meaning postulate:

$$(31) \quad \textit{parent} \text{ N } (x_1 : \langle \text{anim} \rangle (x_1))_{\emptyset} (x_2 : \langle \text{anim} \rangle (x_2))_{\text{Ref}}$$

$$\leftrightarrow$$

$$\textit{child} \text{ N } (x_2)_{\emptyset} (x_1)_{\text{Ref}}$$

This rule says that *parent* and *child* are each other's converses. Obviously, the rule defining converse predicates must be generalized into a general definition such that a statement to the effect that:

$$(32) \quad \textit{converse} (\textit{parent}, \textit{child})$$

is sufficient to establish the desired relationship.

Finally, we must be able to infer, from the proper name *Peter*, that this proper name indicates a male entity. This will be coded in the entry for *Peter*, in some such way as the following:

- (33) Peter $\mathbf{N} (x_1 : \langle \text{male} \rangle (x_1))_{\emptyset}$
 \leftrightarrow
 'Peter' $(d1x_i : \text{name } \mathbf{N} (x_i) : \{(x_1)_{\text{Poss}}\} (x_i))$

That is, to apply the predicate *Peter* to some entity x_1 is to say of that entity, who is male, that the name that he possesses is 'Peter'.

With these various meaning postulates we have the information required to construct, starting from the predication underlying (25a), the predication underlying (25b); that predication can then be used to derive the answer of (24) through the matching operation described in (23).

In the example discussed above, the properties relevant to valid inferencing reside in the lexical elements used and will thus be coded in the lexicon. Obviously, there are many other bits and pieces of linguistic structure relevant to logical inferencing. Let us consider two more examples:

- (34) Knowledge: John bought an expensive book.
 Question: What did John buy?
 Answer: a. An expensive book.
 b. A book.
 c. Something expensive.

Answer (34a) can be simply derived by matching the predications underlying Question and Knowledge. (34b) and (34c) need some inferencing along the lines of:

- (35) If somebody bought an expensive book, then it may be inferred that he bought a book and that he bought something expensive.

In this case the inference is not determined by the lexical meanings of the predicates involved. Rather, it concerns the internal structure of terms, as illustrated in:

- (36 a. Past *buy* $\vee (d1x_i : \text{John } \mathbf{N} (x_i)) \mathbf{A} (i1x_j : \text{book } \mathbf{N} (x_j) : \text{expensive } \mathbf{A} (x_j))_{\text{Go}}$
 b. $(i1x_j : \text{book } \mathbf{N} (x_j))_{\text{Go}}$
 c. $(i1x_j : \text{expensive } \mathbf{A} (x_j))_{\text{Go}}$

This type of inferencing will therefore pertain to Term Logic.

The general structure of terms according to FG may be represented as:

- (37) $(\omega x : \varphi_1 (x) : \varphi_2 (x) : \dots : \varphi_n (x))$

where ω represents the term operators, x the term variable, and each $\varphi (x)$ an 'open predication' acting as a restrictor on the possible values of x . The relevant rule of Term Logic says that from any predication containing a term of the general form (37), we may infer any predication which is otherwise identical, but where the term is specified by any proper subsequence of the restrictors of the original term. This general formulation allows us to draw such inferences as:

- (38) a. John bought an expensive book that he had always wanted to have.
 Therefore:
 b. John bought an expensive book.
 c. John bought a book that he had always wanted to have.
 d. John bought something expensive that he had always wanted to have.

- e. John bought a book.
- f. John bought something expensive.
- g. John bought something that he had always wanted to have.

Therefore, if the Knowledge is as in (38a), an answer to the question *What did John buy?* may sincerely come up with any of (38b) - (38g).

For a last example, consider:

- (39) a. There are six cookies.
 b. John ate two of the cookies.
 c. Bill ate the rest of the cookies.
 Question: How many cookies did Bill eat?
 Answer: Bill ate four cookies.

For the Answer in (39) to be accepted as sincere and correct, we need quite a complicated machinery involving term quantification, elementary arithmetic, and a system for 'anchoring' the referents of terms in the information built up in the context. Much of the machinery required has been sketched in Brown (1985). Informally explained, this machinery works as follows. (39a), which contains the indefinite term:

- (40) $(i6x_i : \text{cookie } N(x_i))$ 'six cookies'

thereby introduces a domain set D whose cardinal is 6. (39b) contains a term with a complex proportional quantifier:

- (41) $(i2/dn x_i : \text{cookie } N(x_i))$ 'two of the cookies'

This term indicates an indefinite subset of cardinal 2 out of a definite set of cardinal n . The definiteness of the superset allows us to identify this set with the domain set D , and to conclude that $n = 6$. (39c) now introduces the term:

- (42) $(d \text{ rest}/dn x_i : \text{cookie } N(x_i))$ 'the rest of the cookies'

where the domain set will again be contextually anchored in D ($n = 6$), and the term operator *rest* will be defined as the complement of R in D , where R is some other subset of D already specified in the context. This must be the subset introduced in (39b). Therefore, the cardinal of 'rest' must be $\text{card}(D) - \text{card}(R) = 6 - 2 = 4$. Therefore, the answer in (39) is correct.

It is to be noted that, given the analysis of term structures in FG, matters of quantification again pertain to Term Logic in FL, rather than to Predicate Logic as is the case in Standard Logic.

6. CONCLUSION AND DISCUSSION

Question-answering can be mediated through simple matching and substitution operations if the following conditions hold:

- (i) Questions are analyzed in terms of underlying FG predications.
- (ii) Answers are constructed from underlying FG predications.
- (iii) Conceptual knowledge is coded in FG predications, plus a Functional Logic which authorizes the derivation of predications which constitute valid inferences from given (sets of) predications.
- (iv) 'Knowing X ' is defined as either possessing a predication ϕ with content X , or possessing a set of predications ψ and a logic FL such that ϕ can be derived from ψ by virtue of FL.

Note the following corollaries to this theory:

1. When I know that Sally is the daughter of Peter, and you know that Peter is the father of Sally, we may, according to (iv), still be said to both know that Peter is Sally's father; however, you know this according to the first alternative of (iv), I know it according to the second one.

2. The *predication* is the central concept in this theory. The predication is (i) the output of the Parser, (ii) the vehicle for knowledge representation, (iii) the input and the output of the inferencing mechanism, and (iv) the input to the Generator. We may add to this that, due to the typological adequacy of the predication, the predication would also constitute the most appropriate level for translating both linguistic expressions and knowledge representations into other languages (cf. Van der Korst, 1986).

3. If a system is provided with a Parser which can map linguistic expressions onto underlying predications, and if the system's knowledge is coded in predications, then we can 'teach' the system (i.e. change or extend its knowledge) by means of linguistic expressions.

For a further elaboration of the theory sketched in this paper, now from the point of view of the interpretation of linguistic expressions, see Dik (1986).

REFERENCES

- Brown, D. Richard (1985). Term operators. In: A.M. Bolkestein et al. eds., *Predicates and terms in Functional Grammar*. Dordrecht: Foris.
- Dik, Simon C. (1978). *Functional Grammar*. Dordrecht: Foris. (Third edition, 1981.)
- Dik, Simon C. (1986). Linguistically motivated knowledge representation. *Working papers on Functional Grammar* 9. Amsterdam.
- Korst, Bieke van der (1986). *Twelve sentences; a translation procedure in terms of Functional Grammar*. MA Thesis, Institute for General Linguistics, University of Amsterdam.
- Kwee Tjoe-Liong (1979). A68-FG(3): Simon Dik's funktionele grammatica geschreven in Algol68 versie nr. 03. *Publikaties van het Instituut voor Algemene Taalwetenschap* 23. Amsterdam.
- Kwee Tjoe-Liong (1987). A computer model of Functional Grammar. In: *Natural Language Generation. Recent advances in AI, Psychology and Linguistics*. Dordrecht/Boston: Academic Press Publishers.