Tilburg University

On-line interpretation in speech understanding and dialogue systems

Bunt, H.C.

Publication date:
1989

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Bunt, H. C. (1989). *On-line interpretation in speech understanding and dialogue systems*. (ITK research report / Institute for Language Technology and Artificial Intelligence; Vol. 1). Unknown Publisher.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

CBM

CBM
R

8409
1989

8409
989-1

UNIVERSITY

THOLIEKE

UNIVERSITEIT

BRABANT

TR 8409

CBM

153
BURG

ITK

RESEARCH
REPORT

ITK Research Report No. 1
Januari 1989

On-line Interpretation in Speech Understanding and Dialogue Systems

Harry C. Bunt

Slightly revised version of a paper, published in H. Niemann, M. Lang and G. Sagerer (eds.) "Recent Advances in Speech Understanding and Dialogue Systems", Springer Verlag, Berlin 1988.

Institute for Language Technology and Artificial Intelligence ITK
Tilburg University, The Netherlands

1	Introduction	3
2	On-line parsing and the problem of discontinuous constituents	5
2.1	Trees with discontinuities	9
2.2	Adjacency in phrase-structure rules for discontinuous constituents	11
2.3	DPSG and parsing	15
3	Semantics: ambiguity and vagueness	18
3.1	On-line interpretation, ambiguity and vagueness	18
3.2	Vagueness and language-understanding systems	19
3.3	Vagueness in two-level semantics	22
3.4	Ambiguity in two-level semantics	25
4	On-line interpretation and the problem of anaphora	26
4.1	Dynamic Logic and natural language	34
5	Pragmatic interpretation: utterance meaning in dialogue	37
5.1	Information dialogues	37
5.2	Contexts in information dialogues	41
5.3	Communicative acts and context change	44
5.4	Communicative functions in information dialogues	49
6	Towards an integrated dynamic interpretation theory	52
6.1	Extending the dynamic-logic approach	52
6.2	Dynamic logic and utterance meaning	55
7	Conclusion	59

On-line Interpretation in Speech Understanding and Dialogue Systems

Abstract

This paper addresses syntactic, semantic and pragmatic aspects of central concern in the design of on-line language understanding systems.

In the area of syntax, the paper focuses on the phenomenon of discontinuous constituents. A form of syntactic representation is defined, called *discontinuous trees*, which allows the constituent structure of sentences with discontinuous constituents to be represented without changing the word order in the sentence. A kind of phrase-structure grammar is defined which can generate such representations, and an on-line parsing algorithm is presented which parses sentences into these representations.

In the area of semantics, the paper focuses on the effective resolution of ambiguity and vagueness, and on the on-line interpretation of anaphora. For dealing with ambiguous and vague expressions in an effective, on-line manner a cascaded model-theoretic approach is developed which makes use of intermediate semantic representations in a formal language that preserves some of the ambiguity and vagueness in natural language. For the interpretation of anaphora, an approach is outlined that has recently been suggested by Groenendijk and Stokhof, based on the use of dynamic logic.

In the area of pragmatics, the paper focuses on the analysis of information-exchange dialogues as consisting of communicative actions. The notion *function of a communicative action* is defined in terms of the flow of information between speaker and addressee. It is argued that Groenendijk and Stokhof's dynamic approach to semantic interpretation can be elegantly combined with this 'functional' approach to communicative action into an integrated theory of utterance meaning. The resulting '*dynamic interpretation theory*' is outlined, which offers exciting perspectives for a full-fledged, on-line utterance interpretation process.

1 Introduction

This paper is concerned with syntactic, semantic and pragmatic aspects of language understanding which are essential to the design of speech understanding and dialogue systems.

A central issue in speech understanding is *on-line* interpretation, i.e. dealing with linguistic information which is relevant to its interpretation as it becomes available, rather than waiting until all the input has been received. The relevant linguistic information includes morphosyntactic, semantic, and pragmatic aspects. In each of these areas one or more crucial problems and some new ideas for their solution will be discussed. Since the syntactic, semantic and pragmatic analysis should go hand in hand in an effective interpretation system, their integration is of the greatest importance. Separate attention will therefore be given to the consequences of the individual problems and solutions discussed for the integration of syntactic processing with semantic and pragmatic processing, and of the integration of semantic and pragmatic processing.

A fundamental problem in on-line processing is that of syntactic (and semantic) *discontinuity*: a syntactic unit may consist of non-adjacent parts. This has the effect that, when the first part of a discontinuous unit is encountered, its contribution to the syntactic (and semantic) interpretation cannot fully be taken into account until the other parts have been encountered. If the material following the first part of the discontinuous unit is to be processed on-line, it follows that we should construct intermediate syntactic representations of a different kind than the familiar constituent-structure trees used in linguistics. I will provide precise definitions of the required syntactic representations, describe a kind of phrase-structure grammar to generate them, and present an algorithm to parse them in an on-line fashion.

Two fundamental semantic problems in on-line processing, which are separated by a somewhat fuzzy boundary, are *ambiguity* and *vagueness*. In the case of an ambiguity, the semantic interpretation process can go several ways to construct semantic representations; in cases of vagueness, the interpretation process lacks sufficient information to go anywhere at all (or, to put it differently, it can go in a virtually infinite number of directions) in constructing a precise formal representation.

Both ambiguity and vagueness can often be resolved using semantic information that becomes available later in the input. But in an on-line process we want to deal with ambiguous and vague elements as they come along. This means that we should construct intermediate semantic representations with a certain amount of ambiguity and vagueness, which get resolved at a later stage. I will describe a cascaded model-theoretic method designed to deal with these problems, which has been used in the PHLIQA question-answering system (Bronnenberg et al., 1980), in the TENDUM dialogue system (Bunt et al., 1985), and in the SPICOS speech communication system (van Deemter et al., 1986; Thurmair, 1988).

A quite different semantic problem concerns the interpretation of anaphoric expressions, which occur very frequently in natural spoken dialogues. The best developed framework for handling anaphora is Kamp's Discourse Representation Theory, in which semantic representations are built up in an on-line fashion as a stretch of discourse is interpreted. This approach is indeed on-line, but it violates the basic principle of semantics, the principle of compositionality. Recently, a new approach to the interpretation of anaphora has been suggested by Groenendijk and Stokhof in terms of dynamic models, borrowing concepts from the theory of programming languages (Groenendijk & Stokhof, 1987). I will describe and discuss this approach, which is claimed to be both on-line and compositional.

Where semantics is the study of those aspects of meaning that concern the relation between linguistic elements and concepts in the universe of discourse, other aspects of meaning which are of particular importance for language understanding in dialogue systems concern the relations between linguistic elements and the beliefs, intentions, expectations, etc. of the speaker. The study of these aspects belongs to the area of pragmatics.

For dealing with these aspects of meaning in the setting of information-exchange dialogues, I have developed an approach where the meaning of an utterance is viewed as the way in which a listener's state of beliefs and intentions is changed as the result of interpreting the utterance (Bunt, 1988a). I will describe this approach, formalizing the concept of the *communicative function* of an utterance as a mathematical function operating on knowledge representation structures.

By combining the pragmatic theory where the 'functional' meaning of utter-

ances is treated in terms of state changes in an interpreter, and Groenendijk & Stokhof's semantic theory where sentence meaning is approached in terms of dynamic models of the universe of discourse, a new, integrated approach to language understanding emerges where all aspects of meaning are accounted for as changing in complex dynamic models. The models in this 'dynamic interpretation theory' are rich and modular knowledge representation structures, describing the states of both participants in the communication.

I will describe this new, integrated approach to language understanding and discuss its perspectives for the possibilities of on-line interpretation in speech understanding and dialogue systems.

Before embarking, let us briefly consider what is meant by an 'on-line' process. A process can have the property of being 'on-line' if it operates on sequential inputs. It has that property if it processes each element in an input sequence immediately, i.e. without waiting for further input elements; the only input information which the process uses is the last input element and the result of processing the preceding input elements. We can express this formally as follows:

- (1) A process F is *on-line* if there is a function f such that
 for every input sequence $\langle i_1, i_2, \dots, i_k \rangle$:

$$F(\langle i_1, i_2, \dots, i_k \rangle) = f(F(\langle i_1, i_2, \dots, i_{k-1} \rangle), i_k)$$

Note that, if $\langle i_1, i_2, \dots, i_k \rangle$ is a sequence of inputs which forms part of a complete input sequence $\langle i_1, i_2, \dots, i_n \rangle$, according to this definition the processing of that part indeed takes place immediately, as it does not depend on later inputs.

2 On-line parsing and the problem of discontinuous constituents

In English, German, French, Dutch and many other languages it happens quite often that the parts of a constituent are not adjacent. In such a case we speak of a *discontinuous constituent*. This phenomenon occurs in a wide variety of constituent types: noun phrases, verb phrases, adjective phrases, adverb phrases, etc. The examples (2) - (9) illustrate this: in (2) we see a discontinuous verb phrase: ('talked about politics'); in (3) a discontinuous preposition phrase ('from which

children'), in (4) a discontinuous adjective phrase ('better than I expected'); in (5) a split verb; in (6) a split negation ('ne ... pas') and a discontinuous noun phrase with 'extraposed' relative clause; in (7) a discontinuous adverb phrase ('harder dan ooit tevoren'); in (8) a discontinuous noun phrases with 'extraposed' prepositional phrase, and in (9) an example of the famous 'cross-serial dependency' construction in Dutch.

- (2) John *talked*, of course, *about politics*
- (3) *Which children* did Anne expect to get a present *from*?
- (4) This was a *better* movie *than I expected*
- (5) *Wake me up* at seven thirty
- (6) Tu *ne crois sûrement pas* qu'un *de tes cousins* viendra *qui demeure à Bruzelles?*
(= Surely you don't think that one of your cousins will come who lives in Brussels?)
- (7) Leo is *harder* gegaan *dan ooit tevoren*
(= Leo has been going faster than ever before)
- (8) Ich habe *ein Auto* gekauft *mit 5 Türe*
(= I have bought a car with 5 doors)
- (9) Ik hoor dat Jan Marie de kinderen de hond heeft helpen leren uitlaten
(= I hear that John has helped Mary to teach the kids to walk the dog)

These examples do not represent a single class of linguistic phenomena, and it is doubtful whether they should all be handled by means of the same techniques.

Sentence (2), which has been discussed extensively in the literature, presents a problem for any analysis in terms of adjacent constituents, since the parenthetical clause 'of course' divides the verb phrase 'talked about politics' into two nonadjacent parts. This means that we are forced to either consider the parenthetical as part of the VP, as Ross (1973) has suggested, or as a constituent at sentence level, as has been suggested by Emonds (1976; 1979). In the latter case, the sentence is analysed as consisting of the embedded sentence 'John talked', with 'of course' and 'about politics' both as specifiers at sentence level. McCawley (1982) provides detailed arguments showing that both suggestions are inadequate (which seems intuitively obvious, from a semantic point of view), and suggests, instead, the syntactic representation (10). This is of course no longer an ordinary tree structure, but McCawley suggests that there is in principle no reason to be afraid

of such constituent structures. We will return to this suggestion below.

(10)

Example (3) represents a different class of phenomena, which are conveniently thought of in terms of movements of parts of phrases. In this example, the NP 'which children' can be thought of as having moved out of the PP 'from which children', of which only the preposition has been left behind. The use of transformations, such as movements, is computationally unattractive, however. In order to deal with such cases without transformations, in GPSG a special type of syntactic categories has been introduced, called 'slash categories'. For instance, the category PP/NP is assigned to a prepositional phrase which 'misses' an NP. In the present example, this category would be assigned to 'from'. The assumption that an NP is missing, propagates to higher nodes in the syntactic tree which the phrase-structure rules construct for the sentence, until it is acknowledged at the top level. Diagram (11) illustrates this.

The technique of using phrases that miss some constituent cannot be used for at least some of the examples (4) - (9), such as (6) and (8). In both sentences the discontinuous NP contains a fullfledged NP, which cannot sensibly be said to 'miss' the relative clause or prepositional phrase that occurs later in the sentence.

(11)

The most straightforward way of dealing with these cases would be to allow syntactic structures with crossing branches as in (10), representing discontinuous constituents as they occur in a sentence, and to formulate grammar rules which describe the discontinuities, producing a structure like (12) for example (8).

(12)

The generation of a representation like (12) seems computationally more attractive than the application of a movement transformation, for the latter approach requires 3 steps: 1. the identification of a discontinuity; 2. the application of the transformation to resolve the discontinuity; 3. the generation of the tree structure for the transformed expression. The generation of (12), on the other hand, can simply be done without the second of these steps.

Whatever techniques may be invented to deal with discontinuities, it seems obvious that a grammar which recognizes discontinuities and describes them in syntactic representations is the most straightforward way of dealing with them, and produces syntactic representations which are the most suitable basis for semantic interpretation.

It therefore seems worth investigating the viability of tree-like structures with discontinuities, like (10) and (12).

2.1 Trees with discontinuities

If we want to represent the situation that a phrase P has constituents A and C, while there is an intervening phrase B, we must allow the node corresponding to P to dominate the A and C nodes without dominating the B, even though this node is located between the A and C nodes:

This will have the consequence that our structures will get crossing branches, if we still want all nodes to be connected to the top node. The above structures (11) and (12) illustrate this. In what respects exactly do these structures differ from ordinary trees? McCawley (1982) has tried to answer this question, suggesting a formal definition for trees with discontinuities by amending the definition of an ordinary tree.

A tree is often defined as a set of elements, called 'nodes', on which two relations are defined, *immediate dominance* (D) and *linear precedence*, or left-right order ($\prec$), which are required to have certain properties to the effect that a tree has exactly one root node, which dominates every other node (immediately or indirectly); that every node in a tree has exactly one 'mother' node, etc. (see e.g. Wall, 1972).

Given the relations of immediate dominance and linear precedence, *dominance* is defined as the reflexive and transitive closure D' of D, and *adjacency* as linear precedence without intervening nodes.

A node in a tree is called *terminal* if it does not dominate any other node; the terminal nodes in a tree are totally ordered by the $\prec$ relation. For nonterminal nodes the precedence relation satisfies the requirement that $x \prec y$ if and only if every node dominated by x precedes every node dominated by y . Formally:

- (14) for any two nodes x and y in the node set of a tree, $x \prec y$ if and only if for all nodes u and v , if x dominates u and y dominates v , then $u \prec v$.

Part of the definition of a tree is also the stipulation that any two nodes either dominate or precede one another:

- (15) for any two nodes x and y in the node set of a tree, either $x D' y$, or $y D' x$, or $x \prec y$, or $y \prec x$.

This stipulation has the effect of excluding discontinuities in a tree, for suppose a node x would dominate nodes y and z without having a dominance relation with node w , where $y \prec w \prec z$. Since neither x dominates w nor w dominates x , by (15) either $x \prec w$ or $w \prec x$. But x dominates a node to the right of w , so by (14) x does not precede w ; and w is to the right of a node dominated by x , so w does not precede x either.

McCawley's definition of trees with discontinuities comes down to dropping the condition that any two nodes should either dominate one another or have a left-right relation. Instead, he proposes the weaker condition that a node has no precedence relation to any node that it dominates:

- (16) for any two nodes x and y in the node set of a tree, if $x D' y$ then neither $x \prec y$ nor $y \prec x$.

We shall call any node u , situated between daughters of a node x without being dominated by x , *internal context* of x . McCawley's definition of trees with discontinuities is inaccurate in some respects; however, his general idea is correct: trees with discontinuities can be defined essentially by relaxing condition (15) in the definition of a tree.

This is only the beginning of what needs to be done, however. The next question is how discontinuous trees can be produced by phrase-structure rules. This question, not addressed by McCawley, is far from trivial and turns out to have interesting consequences for the notion of adjacency in discontinuous trees.

2.2 Adjacency in phrase-structure rules for discontinuous constituents

A phrase-structure rule rewrites a constituent into a sequence of pairwise adjacent constituents. The formulation of phrase-structure rules for discontinuous trees thus requires a notion of adjacency in discontinuous trees. Since we already have precedence relation for discontinuous trees, the obvious definition would seem to be that two nodes are adjacent if one precedes the other, and there are no intervening nodes. This is expressed in (17):

- (17) two nodes x and y in the node set of a tree are adjacent if
and only if $x < y$ and there is no z such that $x < z < y$.

We shall write ' $x + y$ ' to indicate that x and y are adjacent (or 'neighbours'). Unfortunately, this notion of adjacency is of no help for formulating rules that could do anything with internal context constituents. The following example illustrates this. Suppose we are to generate the discontinuous tree structure (18):

To generate the top node, we need a rule combining the V and the NP, like (19):

$$(19) \quad VP \rightarrow V + NP$$

Since the V dominates nodes at either side of the NP, however, there is no precedence relation between the NP and V nodes, leave alone an adjacency relation.

This can be remedied by replacing clause (15) in the definition of a tree by the more general clause (20):

- (20) for any two nodes x, y in the node set of a tree, $x \prec y$ if and only if x 's leftmost daughter precedes y 's leftmost daughter.

(We refrain here from a formal definition of 'leftmost daughter' node, which is intuitively obvious; a terminal node should be understood to be its own leftmost daughter.)

Note that (20) is indeed a generalization of the usual notion of precedence in trees, which is a special case of (20). The recursion in the definition (20) always comes to an end for any finite tree, since the terminal nodes are totally ordered.

Note also that (20) is not consistent with clause (15); by (20), we do get the possibility of a precedence relation between a node and some of its daughter nodes. This is not unreasonable, though. In (21), for example, we do want that $X \prec Y$, and since $Y \prec C$ and the precedence relation

is transitive, we get that $X \prec C$. On the other hand we do *not* want that $X \prec B$, however. We should therefore modify clause (15) to the effect that a mother node does not precede daughter nodes which have no internal context nodes to their left. Formally:

- (22) for any nodes x and z in the node set N of a tree, if $x D z$ and there are no nodes u, v in N such that $x D u$, not $x D v$, and $u \prec v \prec z$, then neither $x \prec z$ nor $z \prec x$.

With the modifications (17) and (22), we have a consistent definition of 'discontinuous trees' which allows us to write phrase-structure rules containing discontinuous constituents as follows:

$$(23) \quad X \rightarrow A + B + [Y] + C$$

where the square brackets indicate that the NP is not dominated by the X node, but is only internal context. The '+' symbol represents the notion of adjacency, defined as before but now on the basis of the revised precedence relation ' $\prec$ '.

- (24) two nodes x and y in a tree are adjacent if and only if
 $x \prec y$ and there is no node z in the tree such that $x \prec z \prec y$.

Upon closer inspection, the neighbour relation defined in this way is unsatisfactory, however, as the following example illustrates.

Suppose we want to generate the following (part of a) tree structure:

To generate the S node, we would like to write a phrase-structure rule that rewrites S into its constituents, like (26):

$$(26) \quad S \rightarrow P + Q + E$$

However, this rule would not apply here, since P , Q and E do not form a sequence of adjacency pairs, as Q and E are not adjacent. According to our definitions, Q has C as its right neighbour, not E . Therefore, the correct rule for generating (25) would be (27):

$$(27) \quad S \rightarrow P + Q + [C] + [D] + E$$

This is ugly, and even uglier rules are required in more complex trees with discon-

tinuities at different levels. Moreover, there seems to be something fundamentally wrong, since the C and D nodes are on the one hand internal context for the S node, according to rule (27), while on the other hand they are also dominated by S . That is, these nodes are *both* 'real' constituents of S and internal context of S .

To remedy this, we introduce a new concept of *adjacency sequence*, which generalizes the traditional notion of (sequence of) adjacency pairs. The definition goes as follows:

- (28) A sequence $\langle a, b, \dots, n \rangle$ is an adjacency sequence if and only if:
1. every pair $\langle i, j \rangle$ in the sequence is either an adjacency pair or is connected by a sequence of adjacency pairs of which all members are a constituent of some element in the subsequence $\langle a, b, \dots, i \rangle$;
 2. the elements in the sequence do not share any constituents.

For example, in the structure (25) the triple $\langle P, Q, E \rangle$ is an adjacency sequence since $\langle P, Q \rangle$ is an adjacency pair and Q and E are connected by the sequence of adjacency pairs $Q - C - D - E$, with C and D constituents of P and Q , respectively. Moreover, P , Q , and E do not share any material. Another example of an adjacency sequence in (25) is the triple $\langle P, B, D \rangle$. The triple $\langle P, B, C \rangle$, on the other hand, is not an adjacency sequence since P and C share the constituent C .

This notion of adjacency sequence can now be used to define a phrase-structure rule for discontinuous trees as a prescription to rewrite a nonterminal into a sequence of constituents which forms an adjacency sequence, where some (but not the first and not the last) of the elements may be marked as internal context. A phrase-structure grammar consisting of rules of this kind we call **Discontinuous Phrase-Structure Grammar**, or **DPSG**.

It is worth emphasizing that this notion of phrase-structure rule is a generalization of the usual notion, since an adjacency sequence as defined by (28) subsumes the usual notion of sequence of adjacency pairs. We have also seen that trees with discontinuities are a generalization of the traditional tree concept. Therefore, phrase-structure rules of the familiar sort coincide with DPSG rules without discontinuous constituents, and they produce the familiar sort of trees without dis-

continuities. In other words, DPSG-rules can simply be *added* to an ordinary PSG (including a generalized or augmented PSG), with the result that the grammar generates trees with discontinuities for sentences with discontinuous constituents, while doing everything else as before.

2.3 DPSG and parsing

A parser for DPSG meets certain complications which do not arise in parsing with an ordinary context-free grammar. To see these complications, we consider what would happen when we apply an ordinary chart parser (see Winograd, 1983) to DPSG.

Context-free chart parsing is a matter of fitting adjoining pieces together in a chart. For example, consider the grammar:

- (30) $S \rightarrow VP NP$
 $NP \rightarrow DET N$
 $VP \rightarrow V$

For the input 'V DET N', a chart parser begins by initializing the chart as follows:

These pieces fit together because the begin- and end vertex numbers of adjoining elements are equal. Given the arc $V(1,2)$ in the chart, we look up all those rules which have a 'free' V as the first constituent. These rules are placed in a separate list, the 'active rule list'. We 'bind' the V 's in these rules to the $V(1,2)$ arc, i.e. we establish links between these rules and that arc. When all constituents in a rule are bound, the rule is applied. In this case, the $VP(1,2)$ will be built, taking over the begin vertex number of its first constituent and the end vertex number of the last. This procedure is repeated for the new VP node. When nothing more can be done, as in this case, we move on in the chart. The final result is the chart (32).

When we use DPSG rules and follow the same procedure, we run into difficulties. Consider the example grammar (33).

$$(33) \quad \begin{aligned} S &\rightarrow VP + NP \\ NP &\rightarrow DET + N \end{aligned}$$

For the input 'V DET N PART' the first constituent that can be built is NP(2,4); the second is VP(1,5). The VP will activate the S rule, but this rule will not be applied since the NP does not have a binding. And even if it had, the rule would not be applicable as the VP(1,5) and the NP(2,4) are not adjoining in the traditional sense.

To make sure that the parser finds all applicable rules of a DPSG, the following provision may be added to the parsing algorithm.

When a rule with internal context is applied, we first follow the standard procedure; subsequently we go through all those rules that appear on the active-rule list as the result of applying the standard procedure and give bindings to those free constituents that correspond in category to the context-element(s) in the rule that was applied.

In the case of (33), this means that just before application of the VP rule (after the PART has been bound), we have the active-rule list (34). (Underlining indicates that a constituent is bound.)

$$(34) \quad \begin{aligned} VP &\rightarrow \underline{V} + [NP] + PART \\ VP &\rightarrow \underline{V} + [\underline{NP}] + PART \\ VP &\rightarrow \underline{V} + [\underline{NP}] + \underline{PART} \end{aligned}$$

We now apply the VP rule. The standard procedure will add one rule to this list, namely $S \rightarrow VP + NP$. The VP is given a binding, so we obtain the following

active-rule list:

- (35) $S \rightarrow \underline{VP} + NP$
 $VP \rightarrow \underline{V} + [NP] + PART$
 $VP \rightarrow \underline{V} + [\underline{NP}] + PART$
 $VP \rightarrow \underline{V} + [NP] + \underline{PART}$

Since the VP rule contained an internal context element, the additional procedure mentioned above is now applied; a binding is given to the NP in (a copy of) the S rule. The S arc is now built in the chart, which does not cause any new active rules. There are no free S's in the old active-rule list either, which would have to be bound, so we look for other rules with a free NP. There is one such rule, the second in (35), but this one will be skipped because it was already present in the rule list before application of the rule; see (34). Skipping of this rule is essential, as there is already a version of the VP rule on the active-rule list containing an NP with the same binding as the context element.

Note that, when applying the rule $S \rightarrow VP(1,5) + NP(2,4)$, we have combined constituents that are not adjacent in the traditional sense, as reflected in successive vertex numbers. In such a case, where the vertex numbers indicate that the constituents in a rule are overlapping, we must perform an additional test whether these constituents form an adjacency sequence.

In order to make sure that only constituents are combined that form an adjacency sequence, the parser keeps track of daughter nodes and internal context in a so-called 'construction list', which is added to each arc in the chart; internal context nodes are marked as such in these lists. With the help of these lists, it is easily detected whether two (or more) nodes share a constituent, in the sense of commonly domination. Moreover, by organizing these lists in a particular way they can also be used to determine whether a sequence of constituents is an adjacency sequence in the sense of definition (28). The reader is referred to Bunt et al. (1987) for a description of the use of construction lists in DPSG parsing.

The important conclusion at this point is that discontinuous syntactic-tree representations of discontinuous constituents can effectively be constructed in an on-line fashion by a slightly modified algorithm for context-free parsing. The importance of this conclusion lies in the fact that such an on-line process produces

syntactic representations that are particularly suitable for semantic interpretation. Methods for systematic semantic interpretation assume the *principle of compositionality*, which is the thesis that the meaning of a sentence is determined by the combination of (a) the meanings of its constituents; (b) the constituent structure. This entails, for a grammar which associates meanings with the expressions of the language, that the syntactic rules should characterize the constituent structure of every expression in a ‘meaningful’ way, using only constituents that are semantically meaningful. Since discontinuous constituents by their very nature form meaningful parts of a sentence, they should be treated as such in syntactic representations. Discontinuous trees offer this possibility, while at the same time representing the linear order of the words in the sentence. Indeed, the use of DPSG allows on-line syntactic parsing integrated with semantic interpretation with the effect that discontinuous constituents are interpreted as they are syntactically recognized. Such an integration has been implemented in the TENDUM system.

3 Semantics: ambiguity and vagueness

3.1 On-line interpretation, ambiguity and vagueness

In this section we consider two difficulties in the formal semantic interpretation of natural-language expressions, which pose a particular challenge for on-line interpretation: ambiguity and vagueness.

Ambiguity is the phenomenon that an expression has several clearly distinct interpretations; vagueness is the phenomenon that an expression has a wide variety of readings, often without clear demarcations. Both ambiguity and vagueness arise at the level of words, as well as at the level of sentences and texts, and they also have in common that their resolution often requires information that becomes available at a later moment in the sentence or text than where they occur. This is the reason why they pose a particular challenge to an on-line interpretation process: an ambiguous or vague expression can often not be interpreted adequately at the moment that it is encountered. Therefore, an on-line process should record the relevant information at that moment, and proceed in such a way that this information can be used at a later stage when more information becomes available.

The latter approach is quite common in the case of *syntactic* ambiguities. For

instance, as we have seen in the previous section, a breadth-first chart parser records the relevant syntactic information in a chart (and, in a sense, in the active rule list), where it is available for further processing at a later stage. Such a parser is in fact facing local syntactic ambiguities all the time. But *semantic* ambiguities, such as lexical ones, are not the business of such a parser. If lexical ambiguities are represented as alternative lexical insertion rules, as linguistic theories have it, the parser has to duplicate all the syntactic work.

For dealing with lexical items which are vague, rather than ambiguous, the situation is in fact still worse. Linguistic semantic theory tells us that vague lexical items are not properly viewed as homonymous words, and hence cannot be handled adequately by means of alternative lexical insertion rules. Instead, a vague lexical item should be treated as a single item whose precise meaning is determined by the surrounding context. But, as long as no methods are provided for how surrounding context determines meaning, this only tells us what a language understanding system should *not* do.

3.2 Vagueness and language-understanding systems

Vagueness in natural language comes primarily from the indeterminacy of lexical items. Consider, for instance, the following ways of using the verb 'have':

- (36) Mary has blue eyes
- (37) KLM has an office in Frankfurt
- (38) Harald has 4 children
- (39) Rosemary is having a baby
- (40) Fidel is having a smoke
- (41) Peter has a cold
- (42) Let me have a look
- (43) They mostly have good weather there
- (44) I had Latin and greek at school

'To have', like 'to do', is one of those verbs that has a large number of possible interpretations, and in fact represents a case where one may doubt whether it should be viewed as ambiguity or as vagueness. Many prepositions, like 'from', 'of', 'in' and 'for' can be used in so many ways, which all have something to do with each other, that it seems hopeless to try and list all the possible interpretations; they represent obvious, problematic cases of vagueness.

The general problem of dealing with vagueness in a semantic theory poses itself very strongly in the context of language-understanding computer systems, due to the fact that such systems mostly require an unusual precision in the interpretation. The following example illustrates this. Webster's dictionary tells us that the word 'Canadian' means: 'pertaining to Canada'. Let us look at some sentences with this word:

- (45) This is a Canadian song.
- (46) I have a Canadian friend.
- (47) Kakabekafalls is a Canadian town.
- (48) The Canadian oak differs from the European oak.
- (49) That's the Canadian flag.
- (50) The Canadian ice hockey team was defeated by the Swedes.
- (51) A Canadian airplane crashed last night.

Clearly, in each of these examples 'Canadian' means something more specific than 'pertaining to Canada'; it means such things as 'made in Canada', 'of Canadian nationality', 'situated in Canada', 'representing Canada', etc. Lexicographers can certainly be excused for not spelling out all cases, however; it seems in fact quite doubtful whether this would at all be feasible.

Now in the case of a language-understanding system, such as a question-answering system, it is necessary to assign very detailed interpretations to such a word. For example, a system which contains information about air planes, like TENDUM, will have to distinguish several interpretations of an expression like 'Canadian air plane':

- (52) air plane manufactured in Canada
- (53) air plane owned by a Canada-based company
- (54) air plane which departed from a city in Canada

The necessity of distinguishing these interpretations can be seen from the following sentences, where in each case at least one of the possibilities should be excluded in view of the surrounding context:

- (55) Does Canadian Pacific only use Canadian air planes?
- (56) Is there a Canadian air plane leaving from Toronto tonight?

In order to determine the intended interpretations of these questions, it is nec-

essary to distinguish between the meanings of 'Canadian' in these sentences. In particular, in order to answer these questions appropriately, a question-answering system will have to be able to figure out how the word 'Canadian' is to be interpreted in terms of the objects and relations in its data base.

It may be noted that in the questions (55) and (56) the adjective 'Canadian' could roughly be paraphrased by the prepositional phrase 'from Canada'. The preposition 'from' presents a particularly striking example of the unusual precision in semantic interpretation required by a question-answering system. Consider the questions (57) and (58):

(57) Does the KL402 come from Canada?

(58) Does the KL402 come from Calgary?

Given a data base where flights have a city of departure, and where cities are known to be located in a country, (57) has to be interpreted as something like (59), and (58) as (60):

(59) $Country(Departureplace(kl402)) = canada$

(60) $Departureplace(kl402) = calgary$

where *Country* and *Departureplace* are data base attributes of cities and flights, respectively. This shows that we are forced to make a distinction between an interpretation of 'from' as 'from a country' and as 'from a city'. In fact, for any two types of objects A and B where we can say 'A from B', we have to distinguish a separate interpretation.

In question-answering systems and other language-understanding systems which require precise semantic interpretation, the commonly found approach is that vagueness is treated as ambiguity, distinguishing a number of alternative interpretations as distinct lexical items. For instance, in one of the early language-processing systems, Raphael (1965) treats 'have' as ambiguous between 'have-as-part', 'have-as-property' and 'own'. This is reasonable from a linguistic point of view, as it may be argued that, generally, 'to have' allows these interpretations. One can see from the examples (36) - (44) that this is clearly not an adequate solution, however, and for application-dependent cases like the precise interpretations of 'from' (and similarly of 'to have') such an approach is obviously not feasible for a nontrivial domain of application. The commonly found 'solution' here is

to distinguish those interpretations which the application requires, and treat the word in the lexicon as ambiguous between those interpretations. This is most unsatisfactory, of course, as it introduces a high degree of application-dependence in the lexicon, presents an ambiguity explosion for the parser, for which separate solutions have to be found, and, from a linguistic point of view, it does exactly what the theory says one should *not* do.

From a theoretical point of view, inherent to the phenomenon of semantic vagueness is that it is impossible to list all the precise interpretations of a vague item. The above example of 'from' confirms this. Therefore, the 'resolution' of the vagueness of such an item should be accomplished by combining semantic information from the context with the general interpretation of the item, which should be vague. However, this presents a fundamental problem both for modern semantic theories, like Montague Grammar or Discourse Representation Theory, and for language-understanding systems, since in both cases the semantic interpretation of natural language takes the form of translating sentences into representation structures, whose semantics is defined relative to a model of the discourse domain. These representation structures aim explicitly at *avoiding* the vagueness and ambiguity of natural language; therefore, these approaches have no place for representations that reflect unresolved vagueness.

A solution to this dilemma has been found by the designers of the PHLIQA system (Bronnenberg et al., 1980), which consists of allowing ambiguous constants in formal semantic representations. Instead of assigning a single denotation to each constant, a set of alternative denotations is assigned. How this works is outlined in the next subsection.

3.3 Vagueness in two-level semantics

The use of ambiguous terms in formal representations to represent lexical vagueness immediately raises the question as to why this would be any better than treating vague words as ambiguous in the lexicon. The answer lies in the use of two (or more) levels of semantic representation. We describe the method in the form in which it has been implemented in the TENDUM dialogue system (Bunt et al., 1985).

Natural-language expressions are interpreted in two steps, each of which re-

sult in a representation in a formal language. The formal languages used for this purpose are two members of the same family, that of Ensemble Languages (the EL family). These languages are based in their semantics on *ensemble theory*, an extension of classical set theory designed for the purpose of natural language interpretation (Bunt, 1985). The languages of the family all have the same syntactic constructions, but differ in their constants. The language used in the first stage, called *EL/F* ('*Ensemble Language, Formal*') has constants that have a one-to-one correspondence with content words (lexemes) in the natural language; one might as well say that the semantic representations constructed at this level are *in terms of the content words*. What the formal language does is making the *formal*, or 'logical' semantic aspects of a sentence explicit - hence the name '*Ensemble Language, Formal*'. The language used in the second stage has constants standing in a one-to-one relation to the elements in a model of the discourse domain. The semantic representations constructed at this level are elaborations of those constructed in *EL/F*, the elaborations consisting in the replacement of the content words by expressions made up of terms that refer to elements in the discourse domain; this language is therefore called '*Ensemble Language, Referential*', or *EL/R*.

The following example illustrates the working of the two levels. Consider the question:

(61) Does the KL402 come from the U.S.?

At the first level of analysis, a formal semantic structure is assigned to the sentence, expressed in *EL/F*:

(62) *COMEFROM*(< *kl402, usa* >)

Here, '*COMEFROM*' is a 2-place predicate of *EL/F*, corresponding to the verb 'come from'; '*kl402*' and '*usa*' are individual constants corresponding to the proper names 'KL402' and 'U.S.', respectively. The second stage of analysis is extremely simple in this case, since the flight 'KL402' and the country 'USA' are simple concepts in the domain of discourse; only 'come from' is not such a concept. Therefore, all that needs to be done is to express the latter relation in terms of *EL/R* concepts, and to replace the *EL/F* constants '*kl402*' and '*canada*' by their *EL/R* counterparts (which are written in capitals, for clarity). The result is:

$$(63) \text{ Country}(\text{Departureplace}(KL402)) = USA$$

Here, 'Country' and 'Departureplace' are EL/F function constants relating a city to its country and a flight to its city of departure, respectively. The second stage of analysis, that of EL/F - EL/R translation, consists simply of looking up the EL/F constants in the EL/F - EL/R lexicon, making appropriate replacements, and simplifying the resulting expression if possible. For the present example, the following part of the lexicon would be relevant:

$$(64) \begin{array}{ll} \text{EL/F:} & \text{EL/R:} \\ kl402 & KL402 \\ usa & USA \\ \text{COMEFROM} & (\lambda x : \text{Departureplace}(\text{elem}_1(x)) = \text{elem}_2(x)) \\ & (\lambda x : \text{Country}(\text{Departureplace}(\text{elem}_1(x))) = \text{elem}_2(x)) \end{array}$$

The variable x in the translations of COMEFROM ranges over pairs of individuals, $\text{elem}_1(x)$ denoting the first element of such a pair, $\text{elem}_2(x)$ the second.

Since the EL/F constants are in fact simply lexical items of the natural language, they carry the same ambiguity and vagueness. As a consequence of this ambiguity, not all combinations of their EL/R interpretations are meaningful. The combinations which are not meaningful are filtered out by means of an articulate type system, which forms part of the EL/R definition. In the example, the first possible translation of COMEFROM would violate the EL/R type restrictions since the type of Departureplace requires the right-hand side of the equality to be an expression of type CITY, whereas USA is of type COUNTRY. Formula (63) is obtained by replacing the EL/F constants in (62) by their translations (for COMEFROM: the second), and simplifying the result by applying lambda conversion.

Having seen, very briefly, how the two-level technique works, we return to the question why the use of ambiguous terms in EL/F for dealing with lexical vagueness would be any better than treating vague words as ambiguous in the lexicon. The answer is that, as we have seen, is it not possible to specify in general all the precise interpretations that a vague word may have. Lexical vagueness can be treated effectively as a case of ambiguity only in connection with a certain domain of discourse. Well, this is precisely what happens in the two-level approach: a vague lexical item of the natural language corresponds with a single term in EL/F; the vagueness gets resolved in the EL/F - EL/R translation, which is the

domain-of-discourse specific part of the semantic analysis. The natural language - EL/F translation, on the other hand, uses only general linguistic information, and a domain-of-discourse independent lexicon.

Clearly, this approach is eminently suited to on-line interpretation. For the natural language - EL/F stage of interpretation, the occurrence of a vague lexical item is nothing special. The entire interpretation process, which translates natural language into EL/R and evaluates EL/R expressions, need not be a sequential two-stage process, but can be designed in such a way that each time a constant is introduced in the EL/F representation under construction, its possible EL/R translations are looked up in the EL/F - EL/R lexicon and attached to the EL/F representation. Information about the types of the EL/R expressions attached to the various constants in the EL/F representation under construction may be used on the fly to rule out some of those expressions. When a complete input sentence has been processed in this way, the possible end results of precise semantic interpretation, i.e. the EL/R representations, can be read from the attachments to the EL/F representations, taking the type constraints into account. (Note that the syntactic structure of the EL/F representations remains unaffected; it is only *expanded* by the EL/F - EL/R translation, if an EL/F constant is translated into a complex expression of EL/R.)

3.4 Ambiguity in two-level semantics

Lexical ambiguity, as opposed to vagueness, can be handled by treating ambiguous words as corresponding to several lexical items; so, for instance, the English noun 'plant' can be represented in the lexicon as plant1: vegetation; plant2: factory; plant3: conspiracy. But if we treat lexical vagueness by means of ambiguous terms in an intermediate level of semantic representation, lexical ambiguity may as well be treated in the same way. This in fact comes down to moving the ambiguity from the natural-language level to the EL/F level and postponing its resolution. This relieves the natural-language parser of the task of ambiguity resolution, and gives rise to a more efficient on-line interpretation process than lexical resolution at an earlier stage would do. Of course, this only applies to *within-category* ambiguity; 'plant' as a verb will have to be distinguished from 'plant' as a noun in the lexicon. More generally, only those cases of lexical ambiguity which have no syntactic repercussions may be postponed.

Where vagueness in natural language has its origins virtually exclusively in the

vagueness of lexical items, ambiguity may have different sources. Besides lexical ambiguity, we have syntactic ambiguity and ‘logical’ ambiguity. Syntactic ambiguities, i.e. different possible interpretations of a phrase as the result of different possibilities to view the phrase as being structured grammatically, are handled in a parser which constructs several alternative syntactic structures in such cases, each of which is interpreted separately by the application of semantic rules. Another type of ambiguity, which we may call *logical*, occurs in the following example:

(65) The boats carried 400 passengers

The ambiguity is whether *each* of the boats under consideration carried 400 passengers, or whether they carried 400 passengers in total. This ambiguity concerns the way the predicate ‘carried 400 passengers’ ‘distributes’ over the set of boats concerned. Such ambiguities may *also* be handled by using ambiguous terms in intermediate EL/F expressions, by introducing special constants for representing distributions (Bunt, 1984). This extended approach to ambiguity procrastination has been implemented in the TENDUM system and taken over in the SPICOS system.

The construction of semantic representations in EL/F is integrated in TENDUM with the use of DPSG, described in the previous section. The complete syntactic/semantic grammar used in TENDUM consists of rules with a syntactic component and a semantic component. The syntactic components are DPSG rules with syntactic categories as complex feature bundles, like in GPSG; the semantic components specify the translation of each syntactic unit into EL/F (see also Thurmair, this volume.)

We thus see that, through the coupling of natural language - EL/F translation rules to DPSG syntax rules, we obtain a parser-interpreter which is able to deal with lexical ambiguity and vagueness as well as with cases of logical ambiguity (and, of course, also with syntactic ambiguity) in an on-line fashion.

4 On-line interpretation and the problem of anaphora

Anaphora are expressions in a discourse (text, dialog, or other) that refer to something that is mentioned elsewhere in the discourse, called its *antecedent*. The

antecedent usually precedes the anaphora, sometimes within the sentence, but often in another sentence. The interpretation of an anaphora thus means that the antecedent be found and in some way be substituted for the anaphora; this is an almost unsurmountable problem for semantic theories that operate within the boundaries of a sentence, as most semantic theories do. But there is an additional problem with anaphora in relation to on-line interpretation. For consider the following two-sentence discourse:

(66) Pedro owns a donkey. He loves it.

The second sentence contains the anaphora 'he' and 'it', referring to Pedro and his donkey, respectively. Now suppose an on-line interpretation process first processes the first sentence, generating a formal representation like (67):

(67) $(\exists x : DONKEY(x) \ \& \ OWN(pedro, x))$

Subsequently the second sentence is processed; assuming that somehow the correct antecedents of the anaphora are chosen, we would expect this sentence to be represented as (68):

(68) $LOVE(pedro, x)$

Since the discourse (66) states that both Pedro owns a donkey and he loves it, we would expect the representation of the discourse to be the conjunction of the representations (67) and (68):

(69) $(\exists x : DONKEY(x) \ \& \ OWN(pedro, x)) \ \& \ LOVE(pedro, x)$

Unfortunately, (69) is not an adequate representation, however, since the variable x in the second conjunct occurs outside the scope of the existential quantifier in the first conjunct; therefore, for the semantics of this formula that second x has nothing to do with the first x ; it is merely an optical illusion that the anaphora 'it' is interpreted as Pedro's donkey. Instead, a correct representation would be (70):

(70) $(\exists x : DONKEY(x) \ \& \ OWN(pedro, x) \ \& \ LOVE(pedro, x))$

To construct this representation, however, we have to break into the representation of the first sentence. This could be done fairly simply for the present example, but for an anaphora whose antecedent is several sentences away, and perhaps hiding in a quantifier within a complex formula, this can be extremely difficult. And even within a single sentence, the correct interpretation of an anaphoric expression may require representations constructed for earlier parts to be revised. This happens in particular in case of reference into a relative clause, or into the if-part of a conditional sentence. The following example illustrates the difficulties.

(71) Every man who loves a woman, admires her.

Here, the problem is that an on-line interpretation process would construct, for the sentence part up to the comma, the representation:

(72) $(\forall x : MAN(x) \rightarrow (\exists y : WOMAN(y) \& LOVES(x,y))\dots)$

Normally, the sentential predicate would be put on the place of the dots, but if we do that here something strange happens:

(73) $(\forall x : MAN(x) \rightarrow (\exists y : WOMAN(y) \& LOVES(x,y)) \rightarrow ADMIRE(x,y))$

The problem is similar to that in (69): the variable y in the $ADMIRE(x,y)$ clause is outside the scope of the quantifier $(\exists y : ..)$, so the anaphoric reference doesn't really succeed. Moreover, the existential quantification is in fact wrong: sentence (71) makes a statement about *any* woman loved by a man, so we should have a universal quantifier instead! We thus see that an expression like 'a woman' should be interpreted as universally quantified in some context, while it should be interpreted as existentially quantified in others.

The severeness of the anaphora problem is brought out by the following figures about the frequency of intersentential anaphora. In a corpus of 960 utterances in human - (simulated) computer dialogues, recorded at the Institute for Perception Research in Eindhoven in a joint project with the Computational Linguistics Unit at Tilburg University, 302 intersentential anaphora were identified, with the following distribution over three major categories:

definite determiner + noun	146	(15%)
definite pronoun	122	(13%)
other	34	(3%)
total	302	(30%)

Keeping in mind that among the 960 utterances is a large number (roughly 50%) of semantically trivial ones, like 'Yes', 'O.K.', 'Good morning', 'Thank you' and 'Goodbye', we see that intersentential anaphora occur more often than not in those sentences that require semantic analysis.

In the early eighties, a breakthrough has been accomplished by Kamp's Discourse Representation Theory (DRT), designed especially to handle anaphora and other intersentential semantic phenomena (Kamp, 1981). DRT interprets texts consisting of more than one sentence in a two-stage process. In the first stage, a semantic representation is constructed in an incremental, 'on-line' fashion. The representation structure which is built for the first sentence in a text is expanded to incorporate the semantic analysis of the second sentence. For a text consisting of several sentences, the end result is one large structure which represents the semantics of the text as a whole; the structure in general does not fall apart into substructures corresponding to the individual sentences. The following examples illustrate the representation structures of DRT, called 'Discourse Representation Structures' (DRSs).

A DRS consists, in the simplest case, of a set of variables, called reference markers, and a set of 'conditions', simple predicate logic formulas. For instance, the sentence (74) has the semantic representation (75), where a predicate constant 'WITP' is used for simplicity to represent the verb phrase 'walks in the park':

(74) A man walks in the park

(75) $\langle \{x\}, \{MAN(x), WITP(x)\} \rangle$

The definition of DRSs is such that (75) is true, given a model M of the domain of discourse, if there is a value assignment to the reference marker x which makes all the conditions true. Therefore, the reference marker is in fact treated as an existentially quantified variable.

When we have a two-sentence text beginning with (74), like (76), the resulting DRS is the one constructed for the first sentence with some additions, depending on the second sentence:

(76) A man walks in the park. He sees Mary.

(77) $\langle \{x, y\}, \{MAN(x), WITP(x), y = Mary, SEE(x, y)\} \rangle$

Here we see the use of reference markers for the treatment of anaphora. For more complex cases, the additions to a DRS constructed earlier may include logical connectives linking one (sub-)DRS to another one, and the definition of these structures ensures that reference markers are treated as existentially or universally quantified. The end result of DRS building is always a structure that is formally equivalent to a predicate logic formula; the distinguishing feature of DRT is thus not the kind of semantic representation that is built, but *the way* it is built.

DRT is the first explicit, formal semantic theory that goes beyond sentence boundaries. Recently, another such theory has been suggested by Groenendijk and Stokhof(1987), which looks particularly attractive from a computational point of view. They propose to use a form of Dynamic Logic, a formalism developed in theoretical computer science for defining the semantics of programming languages (see Harel, 1984).

The starting point of Dynamic Logic is that the meaning of a computer program can be viewed as the way in which the state of a machine changes as the result of executing the program. Of course, the state of a machine after executing program P does not only depend on P but also on the state prior to the execution of P . But we have captured the meaning of P if we specify for every possible initial state the state which results after executing P . Thus, formally, the meaning of a program is a set of machine state pairs $\langle \text{initial state}, \text{final state} \rangle$; 'state transitions', we might say.

This idea relates to the classical way of defining the semantics of formal languages as follows. In the classical approach, interpretation is defined relative to a model M and a function g assigning values to variables. In the simplest case, a model is a pair $\langle D, F \rangle$ where D is the set of individual objects in the domain of discourse and F an assignment of values to the constants of the language. Sometimes it is more convenient to include the variable assignment function g also in the model. We will do so in this paper, and thus consider models of the form (78):

(78) $M = \langle D, F, g \rangle$

The functions F and g , assigning values to constants and variables, are called

the *constant-* and *variable interpretation*, respectively. A formula of the language in question is now true if the values assigned to its constants and variables by F and g are such that the corresponding entities in the domain of discourse have such properties and relations as stated in the formula.

In the Dynamic Logic approach, a computer program is treated as a formal-language expression which contains certain variables, whose values characterize the state of a machine. Execution of the program changes the state of the machine as the values of the variables change. Thus, if g is the function describing the values of the variables before execution of the program, and h the function describing this afterwards, then the pair $\langle g, h \rangle$ represents a state transition that forms part of the meaning of the program.

For example, consider the semantic definition of an assignment statement ' $x := a$ ', where ' a ' may be a complex expression. Its meaning is that, whatever the initial state is, after executing the assignment x has the value of a . This is expressed in (79), where $\|.\|$ stands for 'the meaning of'.

$$(79) \quad \|x := a\| = \{ \langle g, h \rangle \mid h \text{ is equal to } g \text{ except, possibly, for the value assigned to } x, \text{ and } h(x) = \|a\| \}$$

Groenendijk & Stokhof (1987), henceforth referred to as 'G & S', have suggested the possibility of approaching the semantics of formal logical language, like predicate logic, in the same spirit. The meaning of a predicate logic sentence would be described by the set of state transitions taking place in the system (human or electronic) interpreting that formula. For example, the meaning of an existentially quantified statement ' $\exists x : P(x)$ ' would be expressed by (80), where the notation $h[x]g$ is used to indicate that h is equal to g except, possibly, for the value assigned to x :

$$(80) \quad \|\exists x : P(x)\| = \{ \langle g, h \rangle \mid h[x]g \ \& \ h(x) \in F(P) \}$$

(with F as in (78)). Accordingly, the meaning of a predicate-logic sentence is no longer the conditions for its truth or falsity; we are thus leaving the classical idea of truthconditional semantics here. Instead, on this approach the truth value of a formula is derived from its meaning as follows:

- (81) X is true in model $M = \langle D, F, g \rangle$ if and only if there is a value assignment h such that $\langle g, h \rangle \in ||X||$

The following example may help to clarify this.

Consider a model $M_1 = \langle D_1, F_1, g_1 \rangle$ with a domain of 4 individuals: $D_1 = \{a, e, j, m\}$, denoted in DPL as 'adam', 'eve', 'john' and 'mary', two of which are sleeping. In terms of the constant interpretation:

$$(82) \quad F_1 = \{ \langle \text{adam}, a \rangle, \langle \text{eve}, e \rangle, \langle \text{john}, j \rangle, \langle \text{mary}, m \rangle, \\ \langle SLEEP, \{ \langle j, m \rangle \} \} \}$$

Let a state of the model be described by the following assignment of values to variables:

$$(83) \quad g_1 = \{ \langle x, e \rangle, \langle y, a \rangle, \langle w, j \rangle, \langle z, m \rangle \}$$

We consider the meaning of the expression ' $\exists x : SLEEP(x)$ ' and its truth value in the model M_1 . By (80), the meaning of the expression is the set $\{ \langle g, h \rangle \mid h[x]g \ \& \ h(x) \in F(SLEEP) \}$ of state transitions where in the final state the variable x belongs to the extension of the SLEEP predicate. By (81), the expression is true in M_1 if the set

$$(84) \quad \{ \langle g_1, h \rangle \mid h[x]g_1 \ \& \ h(x) \in F_1(SLEEP) \}$$

is not empty. Indeed, it is not, for the pairs $\langle g_1, h_1 \rangle$ and $\langle g_1, h_2 \rangle$ belong to this set:

$$(85) \quad h_1 = \{ \langle x, j \rangle, \langle y, a \rangle, \langle w, j \rangle, \langle z, m \rangle \} \\ h_2 = \{ \langle x, m \rangle, \langle y, a \rangle, \langle w, j \rangle, \langle z, m \rangle \}$$

What this in effect comes down to is that $\exists x : P(x)$ is true in a model M if it is possible to assign to x a value in the extension of P ; this is possible exactly if P has a non-empty extension, which is the traditional condition for this formula to be true. So from the concept of meaning we obtain the traditional truth condition through the definition (81).

By means of rules like (80), G & S present a definition of the semantics of predicate logic; the resulting system they call Dynamic Predicate Logic (DPL). They subsequently propose to use DPL for semantic representation of natural-

language sentences, instead of Discourse Representation Structures, and argue that there are significant advantages in doing so. In the next subsection we consider some of the advantages and limitations of their proposal.

4.1 Dynamic Logic and natural language

G & S claim that the use of DPL is fundamentally better than DRT, since the semantic interpretation process in DRT is not compositional, whereas the use of DPL does give rise to a compositional process. To assess this claim, let us first recall what compositionality means.

A process K is *compositional* if the processing by K of a complex input is a function of (1) the way in which the input is composed of parts; (2) the results of processing the parts. In a compositional interpretation process, each input element is processed separately ('context-free', if it were) and the final result is obtained from these partial results.

We have seen that both DRT and the DPL approach use a two-stage interpretation process 'Int' which can be represented in terms of a process 'R', of constructing formal representations, and a process 'E' of evaluating the representations:

$$(86) \quad Int(\langle i_1, \dots, i_n \rangle) = E(R(\langle i_1, \dots, i_n \rangle))$$

The inputs to R are sequences of linguistic units (words, phrases, sentences,...), from which R constructs representations with a well-defined structuring into parts and specification of the relations between them (like: one is a predicate, another one the first argument of that predicate, etc.) The output of R can thus indeed serve as input to a compositional evaluation process E . Now DRT can be accused of not being compositional, since the DRS-construction stage R is not compositional. The above example (46)-(47) illustrates this: the DRS for the text 'A man walks in the park. He sees Mary.' is not obtained from separate DRSs, constructed for the individual sentences.

To make the comparison with G & S's proposal, let us consider the treatment of anaphora in the DPL-framework. For the text (87a) we would get the DPL-representation (87b):

(87a) A man walks in the park. He whistles.

(87b) $(\exists x : MAN(x) \ \& \ WITP(x)) \ \& \ WHISTLE(x)$

Viewed as a formula in ordinary predicate logic, this would be an inadequate representation since the variable x in $\text{WHISTLE}(x)$ is outside the scope of the existential quantifier. Viewed as a formula in DPL, however, (87) *is* adequate, since G & S define a DPL-conjunction in an order-dependent way as denoting those state transitions which can be obtained by a succession of two transitions, corresponding to the first and second conjunct, respectively. In formula:

$$(88) \quad \|p \& q\| = \{ \langle g, h \rangle \mid \exists k : \langle g, k \rangle \in \|p\| \text{ and } \langle k, h \rangle \in \|q\| \}$$

Combining this with the definition (80) of existential quantification, we see that interpretation of (87) in an initial state g gives rise to an intermediate state k , where an individual is assigned to x that is a man and walks in the park; subsequently, we move to a final state where $\text{WHISTLE}(x)$ is true for the same x . The remarkable thing is that the dynamic semantics of DPL has the effect that free variables in a formula get a value as the result of interpreting some other part of the formula where that variable has a bound occurrence. Intuitively, this is precisely how anaphora seem to work in natural language! This is indeed a very attractive point of DPL.

As DPL, like DRT, offers the possibility of semantic interpretation over sentence boundaries, it is an attractive candidate for use in the interpretation of utterances in a dialogue. However, the DPL framework as proposed by G & S still has some important limitations, which will have to be overcome before it can actually be used for this purpose.

First, an inherent feature of dynamic logic as used in the theory of computer programming, is that the semantic values of the constants of the language are constant; only those of the variables are variable. Consider a computer program consisting of the single statement: $\text{GREATERTHAN}(5,4)$. In dynamic-logic terms, the meaning of this program is the set of those state transitions where nothing happens, but where in the final states the pair $\langle 5, 4 \rangle$ belongs to the extension of the predicate GREATERTHAN . Assuming that this predicate has been defined in the way the name suggests, this is the set of all state changes where indeed nothing happens. (If the predicate has not been defined in such a way that 5 is greater than 4, the meaning of the program is the empty set.) Therefore, the program has the same meaning as, for instance, $\text{GREATERTHAN}(4,3)$ and as $\text{EQUALS}((2+2),4)$. This is clearly undesirable, but rather harmless since programs without any variables are usually uninteresting anyway. Applied to a natural-language sentence

like 'John walks' we get a similar result: given a model where it is the case that John walks, the meaning of the sentence is the set of changes in the model where nothing changes. If, on the other hand, we start with a model where it is not the case that John sleeps, then the meaning of the sentence is the empty set. From the point of view that meaning is the way an interpreter's information changes, this is most unsatisfactory. Also, as in the case of computer programs without variables, no difference in meaning is obtained between sentences whose interpretation does not involve anaphoric or quantified variables, such as 'John walks' and 'Peter sleeps'. In sum, the G & S proposal does not offer a satisfactory way of interpreting expressions made up of constants only.

Second, and closely related to the previous point, is the impossibility of accommodating new information. The dynamic-logic approach to program semantics is in terms of changes in the state of a machine. The state of a machine is considered as always completely determined (although not by the values of the variables in a program alone); in this framework, there is no such thing as a partially determined state. At this point, the parallel between the state of a machine and the information state of a natural-language user goes wrong: one of the main purposes of saying something in natural language is *for informing*, that is, for supplying new information. We will have to consider incomplete information states in order to take this into account.

A third, more technical limitation of the DPL-framework is that the interpretation of bound variables is not quite satisfactory, which creates a problem for the application to natural language (see Bunt, 1988b).

Fourth, and finally, a fundamental limitation has to do with the difference in nature between natural language and formal languages like predicate logic or Prolog. In the DPL-approach, a sentence is interpreted as the set of those changes in the state of an interpreter that make the sentence true. However, a declarative sentence can be used with different functions, such as informing, answering, or asking. If the sentence is used as a question, it should of course be interpreted as a different set of state changes, with final states involving such things as the interpreter knowing that the speaker wants to know whether the sentence is true. The DPL-approach only covers the case where something is asserted, which is in accordance with the use of 'sentences' in languages of logic, but which does not do justice to the use of sentences in natural language. G & S state explicitly as their philosophy:

‘the meaning of a sentence does not lie in its truth conditions, but rather in the way it changes the (representation of the) information of the interpreter. The utterance of a sentence brings us from a certain state of information to another. The meaning of a sentence lies in the way it brings about such a transition.’ (G & S, 1987, p. 1)

Where the shoe pinches is in the distinction between *sentence* and *utterance*. Sentence is a grammatical notion, a string of words with certain syntactic properties. A sentence cannot bring about a change in information state, only an utterance can. It follows, that the state-change approach to meaning should be applied to utterances, rather than to sentences. This means that, rather than considering state changes in an interpreter caused by interpreting a sentence, we should consider state changes caused by a sentence *used as an assertion* or *used as a question*. It also means that we should consider changes in states of much greater complexity than those considered by G & S, who treat the state of an interpreter as simply the set of things he knows (and which are supposed to be true). However, the state of a participant in a dialogue involves at least weak and strong beliefs, goals, and mutual beliefs, as we shall see in the next section. These are the things that change as the result of interpreting an utterance.

5 Pragmatic interpretation: utterance meaning in dialogue

5.1 Information dialogues

By analogy with the notion of an ‘informative question’ as a pure request for information, an ‘informative dialogue’ or ‘information dialogue’ is defined as a case of linguistic interaction in which two partners participate with the sole purpose of accomplishing the transfer of certain information. In its simplest form, such a dialogue consists of an informative question followed by an answer. In real-life situations, an informative dialogue never consists of a question and an answer only; what natural information dialogues look like can be seen in the examples (89) and (90).

Example (89) is a transcription of a telephone conversation with the information service at Schiphol, Amsterdam Airport, simulated in an experiment at

the Institute for Perception Research IPO in Eindhoven in the context of a joint project with the Computational Linguistics Unit at Tilburg University.

(89) S = subject, I = information service

1. I: Schiphol information
2. S: Good afternoon. This is Van I. in Eindhoven. I would like to have some information about flights to Munich. When can I fly there between now and ... next Sunday
3. I: Let me have a look. Just a moment
4. S: Yes
5. I: O.K., there are ... three flights every day, one at nine fifty
6. S: Yes
7. I: one at one forty ... and one at six twenty-five
8. S: Six twenty-five ... These all go to Munich
9. I: These all go to Munich
10. S: And that's on Saturday too
11. I: That's on Saturday too, yes
12. S: Right ... Do you also have information about the connections to Schiphol by train?
13. I: Yes, I do
14. S: Do you know how long the train ride takes to Schiphol?
15. I: You are travelling from Eindhoven?
16. S: That's right
17. I: It's nearly two hours to Amsterdam ... You change there and then it's another fifteen minutes, so you should count on some two and a half hours
18. S: O.K. ... Thank you
19. I: You're welcome
20. S: Bye
21. I: Bye

(The English text given here is a translation from the Dutch original, which is available from the author.)

Dialogue (90) was conducted via computer terminals, it was also recorded at IPO in a joint project with Tilburg University. Of the two participants, one has been trained to act in spoken dialogues like the employees at the telephone information service at Schiphol airport. The other participant (the 'subject') has no experience in the use of computer terminals, and has been told that she participates in an experiment to study the influence of modern technology on information services. The subject has read a little story about a person who plans a holiday trip to Spain and want to know certain details about the possibilities of travelling; the subject was asked to act as being in the position of that person.

(90) (S = subject, I = information service)

1. I: Schiphol information.
2. S: I have booked for flight IB 885 next Saturday to Alicante. What time should I be at Schiphol?
3. I: You should check in half an hour before departure at the latest.
4. S: So between what time and what time?
5. I: Between twelve and one-thirty.
6. S: Do you also have information about departure and arrival times of trains?
7. I: In Holland?
8. S: Yes.
9. I: I do.
10. S: What is the last train from Breda I can take to be there in time for flight next IB 885?
11. I: The train of 12:06.
12. S: What is the arrival time in Alicante?
13. I: 17 hours.
14. S: What is the duration of the bus travel Alicante Benidorm?
15. I: We don't have information about that.
16. S: Thank you.
17. I: You're welcome.

(The English text given here is a translation from the Dutch original, which is available from the author.)

Rather than going into the differences between the spoken and the terminal dialogue (see Beun & Bunt, 1988), we will concentrate here on a few important properties of natural information dialogues in general, which are illustrated in both examples.

First, an information dialogue is much more complex than a sequence of question-answer pairs. In dialogue (90), only 5 of the subject's 9 utterances and 4 of the informant's 9 utterances are factual questions and answers, respectively. In the telephone dialogue (89), there are only 2 factual questions ('When can I fly there between now and next Sunday?' and 'Can you tell me how long the train ride to Schiphol takes?') and two factual answers ('There are three flights every day ..', 'It's nearly two hours to Amsterdam ...'). The other 13 utterances (the exact number depending a little on the way of counting) have to do with verification, acknowledgement, introducing a new topic, or opening and closing the dialogue. We call the latter kind of communicative actions, which, roughly speaking, serve to make sure that the communication proceeds smoothly and stays on the right track, *dialogue control acts*.

Second, it is worth noting that when reading a dialogue text like (89) or (90), one has little problem in identifying the communicative functions of the utterances, be it in informal terms. Exactly how we do this is not immediately clear, but it seems intuitively obvious that the functional interpretation of utterances involves a combination of linguistic features (such as syntactic structure, use of special lexical items, intonation/punctuation) and of the semantic content in relation to the preceding dialogue. There are two points here which deserve further investigation. On the one hand the way in which functions of utterances are recognized, on the other the clarification of the very concept of the (communicative) function of an utterance. The characterization with labels like 'question', 'answer', 'inform', 'verification', 'acknowledgement' is too informal to be immediately useful. For instance, a reply like 'Let me have a look. Just a moment' should perhaps not be called an 'answer', as it does not answer a question. Also, the common intuition is that a verification is a kind of question; we will have to make clear in what sense this is the case. These matters will be taken up in the next subsection.

5.2 Contexts in information dialogues

The study of the functions of natural-language utterances in communication has mostly been undertaken in the borderland of linguistic pragmatics and philosophy.

Unfortunately, even the best developed theories of linguistic communicative action, or 'speech acts', such as those of Searle (1969), Allwood (1976), or Gazdar (1979), do not have the explicitness and formality required for use in building computer dialogues systems. Levinson (1983) argues that the most promising approach for arriving at such a theory would be one where speech acts are characterized in terms of their context-changing effects, as has for instance been proposed by Gazdar (1979), and Bunt (1977). But as the notion of context in general is an extremely broad and vague one, this does not seem to be a feasible approach. I believe, however, that by limiting ourselves to pure information dialogues, we can obtain a notion of context which is sufficiently clear and manageable to provide the basis for an explicit formal characterization of linguistic communicative action. In this subsection I will indicate the structure of these contexts.

Inherent to the very notion of an information dialogue is that we have two partners A and B with the purpose of obtaining or providing factual information. The communicative actions that A and B perform will obviously have effects on their information. So, basic to the notion of context we are looking for, must be what A and B know and in what respects they want to expand their knowledge. However, we must be careful using the term 'knowledge', for two reasons. First, it would perhaps be better to speak of 'belief' rather than 'knowledge', since we do not want to exclude the possibility that a participant has incorrect information. What is meant here by saying that A knows that p, is just that A has the information p available, without implying the actual truth of p. This corresponds to one of the uses of the verb 'to know'; there is another use according to which something which is known must be true. It is the former use which is relevant here, since the course of the dialogue between A and B does not depend on what is actually true, but on what A and B believe to be true.

Second, beliefs can be uncertain. Natural information dialogues contain a substantial amount of verification, as example 2 illustrates. Apparently, participants in information dialogues often have uncertain knowledge about something and a desire to resolve their uncertainties. The effects of communicative actions in information dialogues thus include effects on the uncertain knowledge of the participants. I shall describe the situation where a participant A has some information p available, without fully trusting it, as 'A suspects that p'. So in describing the context of an informative dialogue, two important types of elements we must take into account are:

- (91) A knows that p
- (92) A suspects that p

with the interpretation of 'knows' and 'suspects' discussed above. What 'p' may stand for will be discussed below.

In an information dialogue, not only the information available to the partners is important but also the information which is not available, and in particular the information they want to become available. There are two ways in which one may want information to become available: one may want it to become available to oneself or one may want it to become available to the partner. In other words, one may want to know something or one may want to make something known. These are the only kinds of goal that can underly an information dialogue. As ingredients of the context that gives rise to an information dialogue we must therefore also take into account elements of the form

- (93) A wants to know whether p
- (94) A wants to make known that p.

The things that A and B may want to know or to become known are, in the first place, specific facts about the domain of discourse; that is what the dialogue is all about. However, sometimes a participant may want to know or to make known certain information about oneself or the other. We see examples of this in dialogue (90), where participant S wants to know, in line 6, what information is available to the other; where participant I wants to know for certain in line 7 that the other refers to trains in Holland, and where participant I wants to make known in 15 that she does not have information about buses in Spain. Apparently, we should allow the 'p' in (91) - (94) to be information about the discourse domain as well as about what the speaker knows or what the other knows. Moreover, when we look at line 14 in dialogue (90), we see that a participant may want to know something which cannot be expressed in the form of (93); this form only allows for the case where one wants to know the truth of a proposition. We therefore need a more general form than (93), namely:

- (95) A wants to know the value of x

where x may be a proposition, in which case the value is either TRUE or FALSE, but may also be something else, such as the duration of a train ride, or the time.

This generalization is also required for the 'know' attitude and the 'suspect' attitude: one may know (or suspect) that a plane has flight number KL402, but one may for example also suspect that the partner knows what the flight number is. Apparently, we must distinguish between the following two cases:

- (96) A knows the value of x
- (97) A knows that the value of x is y

Similarly for the 'suspect' attitude. (Although in English one doesn't say 'I suspect the value of ...', it is clearly possible to have information which is not fully trusted about the value of something.) In combination with the observation that the objects of the various attitudes, indicated by 'p' and 'x' in these formulae, may be aspects of the discourse domain as well as aspects of the dialogue partner, this means that we should allow context elements with nested speaker/attitude references, such as:

- (98) A knows that B knows the arrival time of flight KL402
- (99) A knows that B wants to know the arrival time of flight KL402
- (100) A knows that B suspects that the arrival time of flight KL402 is 12:45
- (101) A knows that B suspects that A knows the arrival time of flight KL402

and so on. In sum, what we find as key elements of the context can be described by the following recursive rules:

- (102) 1. [A, B] [knows, suspects, wants to know, wants to make known] the value of x, where x is information about the discourse domain.
- 2. [A, B] [knows, suspects, wants to make known] that p, where p is a proposition expressing information about the discourse domain.
- 3. [A, B] [knows, suspects, wants to know, wants to make known] the value of p, where p is a proposition of the form described by 1, 2, 3 or 4.
- 4. [A, B] [knows, suspects, want to make known] that p, where p is a proposition of the form described by 1, 2, 3 or 4.

These rules then describe the types of beliefs and intentions that are hypothesized to make up the 'context' that gives rise to and is affected by the utterances in an information dialogue. Schematically, we can represent such a context as a pair:

$$(103) \quad K = \langle K_A, K_B \rangle$$

where K_A and K_B are the respective sets of beliefs and intentions of A and B, as characterized by the rules (102). (Note: in Bunt (1988a) contexts are represented as unordered pairs; it turns out to be technically advantageous to order these pairs.)

5.3 Communicative acts and context change

The idea behind exploring the notion of context in the preceding subsection was that different types of communicative action can be distinguished in terms of their effects on the context. Clearly, the effects on a context as defined above consist primarily in additions to the addressee's information. For instance, if speaker A asks speaker B whether flight KL402 comes from Montreal, B is provided with at least the following information:

(104a) A wants to know whether flight KL402 comes from Montreal

(104b) A suspects that B knows whether flight KL402 comes from
Montreal

And if B answers that this is indeed the case, A is provided with at least the following information:

(105) B knows that flight KL402 comes from Montreal

But this is not all, for the performance of a communicative act does not only influence the information of the addressee, but also that of the speaker. For one normally assumes, unless there is evidence to the contrary, that the action one performs does transmit the information which it conveys when correctly understood. For instance, asking whether flight KL402 comes from Montreal not only has the effects (104), on B, but also the effects (106), on A:

- (106a) A suspects that B knows that A wants to know whether flight
KL402 comes from Montreal
- (106b) A suspects that B knows that A suspects that B knows whether
flight KL402 comes from Montreal

Schematically, the effects of a communicative action can thus be represented as follows:

$$(107) \quad K = \langle K_A, K_B \rangle \implies K' = \langle K'_A, K'_B \rangle$$

where K'_A and K'_B are A's and B's respective beliefs and intentions, 'updated' as the result of the action. There is a simple relationship between the effects on the addressee and those on the speaker, as (104) and (106) illustrate, namely that for every effect E_j on the addressee there is a corresponding effect on the speaker of the form

$$(108) \quad A \text{ suspects that } E_j$$

This expresses simply that the speaker (A) 'suspects' his action to be correctly understood. It should be emphasized that the effects considered here are only those which are indissolubly connected to the action. In speech act theory, these effects are often called illocutionary. Of course, an action like answering that the KL402 is leaving in 20 minutes may have other effects than that the addressee knows the speaker to know that, such as the addressee starting running to the gate. Such 'indirect' effects, which are often called perlocutionary, obviously play no part in the characterization of communicative action and are therefore not considered here, important though they are. Note that the 'direct' effects E_j on an addressee B, resulting from the understanding of the communicative act, are always of the general form (109):

$$(109) \quad E_j = B \text{ knows that } C_j$$

where C_j is some condition on the state of the speaker. The above examples illustrate this. The conditions C_j are in fact those conditions that the speaker's state should satisfy in order for him to act in accordance with the implicit rules of the information dialogue game, which include certain general principles con-

cerning cooperativeness, honesty, rationality, etc. (see Allwood, 1976). These are sometimes called the *appropriateness conditions* of the action in question. For a communicative action with appropriateness conditions $\{C_1, \dots, C_k\}$, performed by speaker A and addressed to B, we thus get the following direct effects, inherent to the recognition of the action:

- (110) B knows that C_j
 A suspects that B knows that C_j
 ($j = 1, \dots, k$)

And this is only the tip of an iceberg. For the participants A and B surely both know, in principle, that these effects occur. Therefore, B also knows that A suspects B to know that C_j :

- (111) B knows that A suspects that B knows that C_j

By the same token, A knows, at least in principle, that B knows this:

- (112) A knows that B knows that A suspects that B knows that C_j

And by the same token, B knows, at least in principle, that this is the case:

- (113) B knows that A knows that B knows that A suspects that
 B knows that C_j

And so on. More of this infinite iceberg is shown in (114), where we use the notation ' $A \models p$ ' for 'A knows that p' and ' $A \models\!\!= p$ ' for 'A suspects that p':

$$\begin{array}{rcl}
(114) & & B \models C_j \\
& & A \models B \models C_j \\
& & B \models A \models B \models C_j \\
& & A \models B \models A \models B \models C_j \\
& & B \models A \models B \models A \models B \models C_j \\
& & A \models B \models A \models B \models A \models B \models C_j \\
& & B \models A \models B \models A \models B \models A \models B \models C_j \\
& & A \models B \models A \models B \models A \models B \models A \models B \models C_j \\
& & B \models A \models B \models A \models B \models A \models B \models A \models B \models C_j \\
& & A \models B \models A \models B \models A \models B \models A \models B \models A \models B \models C_j
\end{array}$$

It seems that, in principle, there is no end to the iteration of speaker/hearer pairs (cf. Clark & Marshall, 1982 for a discussion of this phenomenon), though for practical purposes it may of course be reasonable to impose a limit on the recursion somewhere. We should thus extend (110) with (111), (112), (113), and ever deeper nestings of alternate participant/attitude pairs, as (114) indicates. It is instructive to separate the context effects that thus arise into the effects on A and those on B; this gives us (115) and (116):

- (115) 1. B knows that C_j
 2. B knows that A suspects that B knows that C_j
 3. B knows that A knows that B knows that A suspects that B knows that C_j
 etc.
- (116) 1. A suspects that B knows that C_j
 2. A knows that B knows that A suspects that B knows that C_j
 3. A knows that B knows that A knows that B knows that A suspects that B knows that C_j
 etc.

These effects can be represented in a finite form by introducing a 'one-sided mutual knowledge operator'. By *mutual knowledge* of a certain fact f is meant that both A and B know that f , they both know that the other knows that f , they both know that the other knows that the other knows that f , and so on (see e.g. Clark & Marshall, 1981; Perrault & Cohen, 1981). One-sided mutual knowledge is one half of mutual knowledge, so to speak, namely the set of those pieces of mutual

knowledge that constitute the knowledge of one speaker. Using such a one-sided mutual knowledge operator KMK, for '*know to be mutually known*', we can represent the effects (115) and (116) on A and B in finite form:

(117a) KMK(A, B, A suspects that B knows that C_j)

(117b) B knows that C_j & KMK(B, A, A suspects that B knows that C_j)

Mutual knowledge operators can be given a direct interpretation in finite form (Halpern & Moses, 1984) which means that they can be treated as finite primitive concepts rather than as mere abbreviations of infinite notions. This makes their use acceptable from a psycholinguistic point of view, and feasible from a computational point of view.

For the present discussion, the important conclusion is that the direct effects of a communicative act, represented schematically as (107) above, can be represented as follows:

(118) $\langle K_A, K_B \rangle \Rightarrow$
 $\langle \text{updt}(K_A, \{ \text{KMK}(A, B, A \text{ suspects that } B \text{ knows that } C_j(c)) \}),$
 $\text{updt}(K_B, \{ B \text{ knows that } C_j(c) \} \cup$
 $\{ \text{KMK}(B, A, A \text{ suspects that } B \text{ knows that } C_j(c)) \}) \rangle$

Here we have introduced a function *updt* of two arguments: (1) a set of beliefs and intentions, constituting the 'state' of a dialogue participant, and (2) a set of beliefs with which this state has to be 'updated'. In the simplest case these beliefs can be added to those already present in that state, but in general the update has to take the revision of beliefs and intentions into account. We will return to this function in section 6.

In (118) we have written $C_j(c)$, rather than just C_j , in order to indicate that the appropriateness conditions C_j depend on the semantic content c of the communicative act. From (118) we obtain a definition of the concept of communicative function as a context-changing function in the mathematical sense, by performing lambda abstraction over the semantic content and the context. This gives us the following function CF, where A and B are the 'owners' of the states K_A and K_B that make up a context K :

$$(119) \quad CF = (\lambda z, K : \\
\quad < updt(K_A, \{KMK(A, B, A \text{ suspects that } B \text{ knows that } C_j(z))\}), \\
\quad \quad updt(K_B, \{B \text{ knows that } C_j(z)\} \cup \\
\quad \quad \quad \{KMK(B, A, A \text{ suspects that } B \text{ knows that } C_j(z))\}) >$$

The agent whose state is changed in the way indicated at the first use of the *updt* function in (119) plays the role of the speaker, the other that of the addressee. Application of this function to a certain content *c* gives us the communicative act $CA = CF(c)$, construed as a function with the context-changing effect described in (117). This is in line with the classical view of the concept of illocutionary force as a function that can be applied to a propositional content to yield an illocutionary act (Searle, 1969; Gazdar, 1979b). An interesting point to note is that the effects which are characteristic for a communicative act can be derived systematically from its appropriateness conditions. This means that the simplest way of distinguishing one communicative function from another is, after all, by the set of appropriateness conditions.

5.4 Communicative functions in information dialogues

Let us now consider the application of the approach to communicative action, outlined in the previous subsection, to information dialogues. In information dialogues, contexts consist of beliefs and intentions as described by (102), and so do appropriateness conditions and the possible packages of them that correspond to communicative functions.

In the above discussion we have come across three different types of motivation that may underly a dialogue act: (1) the speaker wants to know something; (3) the speaker wants to make something known; (4) the speaker knows that the hearer wants to know something. This trichotomy can be used as a starting point for a taxonomy of communicative functions as falling into three categories: question, inform, and answer functions. For each of the three categories a class of functions by means of the corresponding package of appropriateness conditions has been characterized in Bunt (1988a). Some of these functions are more specific than others, by which we mean that the set of appropriateness conditions of one includes that of another. This is conveniently represented in a hierarchical diagram, like (120), which shows the class of yes/no questions and its instances.

(120)

YES/NO QUESTION FUNCTIONS:

The appropriateness conditions of the various functions contain, of course, the semantic content to which the function is applied, according to (119), to form a dialogue act. For yes/no-questions (as well as for informing functions) the semantic content is always a proposition, designated in (120) by *p*.

A point of interest to consider is the logical independence of the appropriateness conditions within a package corresponding to a communicative function. In the speech act literature it is usually assumed that one of the appropriateness conditions of a question is that the speaker does not possess the information he asks for (a 'sincerity condition'). Such a condition has not been included in (120), in view of the fact that wanting something to be the case is usually understood as implying that this something is not already the case. Therefore, if *S* wants to know whether *p*, it is logically implied that *S* does not know whether *p*. Logically redundant appropriateness conditions have been left out in (120).

It may also be observed that the appropriateness conditions for a CONTRA-CHECK with content p are identical to those of a NEGA-CHECK with content *not* p . The reason for nonetheless distinguishing the two is that p in both cases stands for the semantic content of the utterance, and that a CONTRA-CHECK can be expressed by a positive sentence, as in (121):

(121) Does he come from *Montreal*?

with heavy stress on the last word. Now this sentence would normally be assigned the semantic content *He comes from Montreal*. So in order to interpret the utterance as a NEGA-CHECK with content *He does not come from Montreal*, we would have to use the intonation to change the semantic content. Basically, we can interpret intonation in two ways: semantically or pragmatically. Since intonation in general is one of the major sources of information for determining the communicative function of an utterance, and is therefore interpreted pragmatically, we think it is preferable to interpret the intonation pragmatically here as well. This means that we have the semantic content *He comes from Montreal*, and we assign the utterance a different communicative function: CONTRA-CHECK.

Interestingly, in Dutch and in several other languages it does not seem possible to express the verification of the falsity of a certain proposition, or the verification of suspected agreement concerning the falsity of a certain proposition, without using an explicit negation. This is the cause of the asymmetry in the tree structure dominated by the YN-QUESTION in (120). Note that this is a purely empirical matter.

Other types of questions, such as WH-questions, multiple-choice questions, examination questions and requests, as well as informing and answering functions are defined and discussed in Bunt (1988a).

6 Towards an integrated dynamic interpretation theory

6.1 Extending the dynamic-logic approach

In this section I will indicate how the dynamic-logic approach to semantics and the context-change approach to pragmatics may be combined into an integrated

theory of natural-language interpretation.

Pragmatic interpretation according to the context-change approach described in the previous section is based on the transfer of information between speaker and hearer, including cases where previous beliefs are changed and where gaps in an incomplete state of information are filled. We have seen earlier that two severe limitations of the dynamic-logic approach are precisely its failure to deal with these cases. To overcome these limitations we have to consider extending the dynamic-logic approach in two respects:

- allow not only value assignments to variables to change, but assignments of values to constants as well;
- allow DPL-models to be partial; in particular, allow the assignments of values to variables and constants to be partial functions.

In this subsection we consider such extensions. We first consider the case of changing beliefs.

An obvious consequence of allowing both constant and variable interpretations to change, is that a *state in a model* $M = \langle D, F, g \rangle$ is no longer just a specification of g , but a specification of both F and g : a pair $\langle F, g \rangle$. We consider the example of an interpreter whose information is represented by the state $\langle F, g \rangle$ of a model. For this interpreter we consider the semantic interpretation of the predicate-logic sentence:

$$(122) \quad SLEEP(adam)$$

If we take the meaning of this formula to be the way it changes an interpreter's information, the dynamic interpretation of (122) should bring our interpreter in a state $\langle G, h \rangle$ where it becomes (or remains) true that Adam sleeps. This can be accomplished by extending the extension $F(SLEEP)$ of the predicate $SLEEP$ so as to include the individual denoted by *adam*, if it didn't already do so. The meaning of (122) is the set of all such state changes, for any initial state $\langle F, g \rangle$, so we obtain as the meaning of (122):

$$(123) \quad ||SLEEP(adam)|| = \{ \langle \langle F, g \rangle, \langle G, h \rangle \rangle \mid h = g \ \& \ G[SLEEP(adam)] = F \ \& \ G(adam) \in G(SLEEP) \}$$

We use the notation $G[P(j)]F$ to indicate that G and F are identical except, possibly, for the values they assign to the predicate P , applied to the argument j .

If the information of an interpreter consists not only of the contents of a model but also of postulates which are external to the model, the consequences of the changes in G required by these postulates have to be taken into account as well; see Bunt (1988b). In the rest of this paper, I only consider the case where an interpreter's information is represented in a model without additional constraints.

Next we consider the accomodation of new information to an incomplete state of information. If an incomplete state is to be modelled by pairs $\langle F, g \rangle$, as before, then the functions F and g should be partial, i.e. they should not necessarily assign a value to each of the constants and variables of the DPL language.

The most interesting form of incompleteness concerns the interpretation of predicate constants, in case the extension of a predicate is not completely known. If P is such a predicate constant, $F(P)$ is undefined. However, the extension of P being partly known, we would of course like to say that $F(P)$ is *not completely defined*, rather than undefined. In realistic situations, it will often be the case that certain individuals are known to belong to the extension of a predicate, certain individuals are known not to belong to it, and of certain individuals it is not known whether they belong to it. This cannot be represented in the traditional way by means of a single function F ; instead, we need two functions, say F and F' , where $F(P)$ is the set of individuals known to belong to the extension of P , and $F'(P)$ the set of individuals known not to belong to the extension of P .

This interpretation of F , plus the introduction of the function F' , specifying known 'negative extensions' of predicate constants, has the consequence that F (and F') can be a total function, after all. Where in the traditional approach F would, for instance, be undefined if nothing is known about the extension of a predicate P , we now represent this by $F(P) = F'(P) = \emptyset$.

Partial models are thus not simply models with partial functions, but 4-tuples:

$$(124) \quad M = \langle D, F, F', g \rangle$$

where F and F' are total functions, F having the set of all constants of the DPL language as its domain; F' only the set of predicate constants. Consequently, a state will now be a triple $\langle F, F', g \rangle$ rather than a pair $\langle F, g \rangle$.

As an example, we consider a partial model $M = \langle D, F, F', g \rangle$, with a domain consisting of four individuals: $D = \{a, e, j, m\}$, and the following state

$\langle F, F', g \rangle$, with F, F' and g as in (125):

$$\begin{aligned}
 (125) \quad F &= \{ \langle \text{adam}, a \rangle, \langle \text{eve}, e \rangle, \langle \text{john}, j \rangle, \langle \text{mary}, m \rangle, \\
 &\quad \langle \text{MAN}, \{a, j\} \rangle, \langle \text{WOMAN}, \{e, m\} \rangle, \langle \text{SLEEP}, \{m\} \rangle \} \\
 F' &= \{ \langle \text{MAN}, \{e, m\} \rangle, \langle \text{WOMAN}, \{a, j\} \rangle, \langle \text{SLEEP}, \{e, j\} \rangle \} \\
 g &= \{ \langle x, a \rangle, \langle y, e \rangle, \langle z, j \rangle, \langle w, m \rangle \}
 \end{aligned}$$

There is complete information in this state as to who is a man and who is a woman, but there is incomplete information about who is sleeping and who is not. Mary is known to sleep and Eve and John are known not to sleep, but whether Adam sleeps is not known. We consider once again the interpretation of the DPL statement (122): $\text{SLEEP}(\text{adam})$. As the result of interpreting this statement, an interpreter whose state of information is described by (125) will add the individual a to the extension of SLEEP and remove it from the 'negative extension' $F'(\text{SLEEP})$, if necessary. The meaning of $\text{SLEEP}(\text{adam})$ as expressed in (123) in fact already covers this; we only have to adapt (123) to partial models in order to cover the case that Adam was known not to sleep. We thus get:

$$\begin{aligned}
 (126) \quad ||\text{SLEEP}(\text{adam})|| &= \langle \langle F, F', g \rangle, \langle G, G', h \rangle \rangle \mid h = g \ \& \\
 &\quad G[\text{SLEEP}(\text{adam})]F \quad \& \\
 &\quad G(\text{adam}) \in G(\text{SLEEP}) \quad \& \\
 &\quad G'[\text{SLEEP}(\text{adam})]F' \quad \& \\
 &\quad G(\text{adam}) \notin G'(\text{SLEEP}) \}
 \end{aligned}$$

Let us now consider the introduction of new information by a quantified DPL-sentence, such as $\exists x : \text{MAN}(x) \ \& \ \text{SLEEP}(x)$. Note that, in state (125), no man is sleeping. Straightforward adaptation of the G & S interpretation of existential quantification to partial models would give us the following:

$$\begin{aligned}
 (127) \quad ||\exists x : \text{MAN}(x) \ \& \ \text{SLEEP}(x)|| &= \\
 &= \{ \langle \langle F, F', g \rangle, \langle G, G', h \rangle \rangle \mid F = G \ \& \ F' = G' \ \& \ \exists k : k[x]g \ \& \\
 &\quad \langle k, h \rangle \in ||\text{MAN}(x) \ \& \ \text{SLEEP}(x)|| \} \\
 &= \{ \langle \langle F, F', g \rangle, \langle G, G', h \rangle \rangle \mid F = G \ \& \ F' = G' \ \& \ h[x]g \ \& \\
 &\quad h(x) \in F(\text{MAN}) \ \& \ h(x) \in F(\text{SLEEP}) \}
 \end{aligned}$$

This is inadequate, however, since the constant interpretation functions F and F' are such that no man is in the extension of SLEEP ; therefore, it is not possible that $h(x) \in F(\text{MAN})$ and $h(x) \in F(\text{SLEEP})$. To accomodate the new

information that a man is sleeping, F and F' have to change as well. There are several possibilities: Adam or John can be brought into the extension of *SLEEP*, or Eve or Mary, if they are also moved from the extension of *WOMAN* to that of *MAN*. And of course any combination of these changes would do as well.

We thus obtain the following interpretation:

$$\begin{aligned}
 (128) \quad ||\exists x : MAN(x) \& SLEEP(x)|| = \\
 &= \{ \langle \langle F, F', g \rangle, \langle G, G', h \rangle \rangle \mid h[x]g \& \\
 &\quad G[SLEEP(x), MAN(x)]F \& G'[SLEEP(x), MAN(x)]F' \& \\
 &\quad h(x) \in G(MAN) \& h(x) \in G(SLEEP) \& \\
 &\quad h(x) \notin G'(MAN) \& h(x) \notin G'(SLEEP) \}
 \end{aligned}$$

This still does not seem quite satisfactory, due an unsatisfactory point which is inherited from DPL. According to (128), an interpreter who processes the DPL-sentence asserting that a man sleeps, would get into a state where either he knows that Adam sleeps or that John sleeps (or both, or Eve or Mary, assumed to be a man). But shouldn't the new state rather be that it is known that one of them sleeps, without knowing which one? This requires more complex models. This complication is due to the fact that we now look at states of a model as representing *an interpreter's knowledge*, where states in DPL-models represent states of affairs in a domain of discourse. This shift in the view on states and models closely relates to the shift we want to make from a state-change semantics approach to a state-change approach to the interpretation of utterances.

6.2 Dynamic logic and utterance meaning

As a point of fundamental criticism on the DPL-approach, we have mentioned in section 4 that G & S consider sentence meaning in terms of changes in an interpreter's state of information, but such changes are not accomplished by sentences (either in DPL or in natural language), but by communicative actions: sentences uttered with a certain communicative function.

So let us look at the state-change interpretation of the utterance 'Adam sleeps', for example, addressed to hearer H by speaker S in order to inform H that Adam sleeps. The immediate suggestion would be to use (126) as the interpretation of this utterance rather than of its semantic content. Describing the utterance as a quadruple $\langle \textit{communicative function}, \textit{semantic content}, \textit{speaker}, \textit{addressee} \rangle$ this would give us (129):

$$\begin{aligned}
(129) \quad & || < inform, SLEEP(adam), S, H > || = \\
& \{ < < F, F', g >, < G, G', h > > \mid h = g \ \& \\
& \quad G[SLEEP(adam)]F \ \& \ G'[SLEEP(adam)]F' \ \& \\
& \quad G(adam) \notin G'(SLEEP) \}
\end{aligned}$$

But according to the pragmatic theory outlined in the previous section, this communicative act would have considerably more complex effects than (129) suggests. The effects on the interpreter H would be (130):

- (130) H knows that S wants to make known that Adam sleeps
H knows that S knows that Adam sleeps
H knows that H and S mutually know that S suspects that H
knows that S wants to make known that Adam sleeps
H knows that H and S mutually know that S suspects that
H does not know that Adam sleeps

There would be effects on the state of S too (see (117b)), but we will leave those out of consideration for the moment. The effects (130) should be contrasted with the effect assumed in the DPL approach reflected in (129), which is simply that H knows that Adam sleeps. To take (130) into account, we need more complex state descriptions than the triples $< F, F', g >$. Since we have to take two agents into account and various attitudes toward the information that Adam sleeps, we can no longer get by with a single function F , describing a constant's extension as far as known, and its negative counterpart F' . We need to make the description of extensions relative to speaker and hearer and to their attitudes. It is beyond the scope of the present paper to discuss the structure of the more complex models to which this leads (Bunt, in prep.), but we will indicate here what we get by assuming that our models M contain sets of interpretation functions which assign (partial) extensions to predicate constants relative to S's knowledge, to H's knowledge, to S's knowledge of H's knowledge, to S's suspicions, etc. The various F functions will be distinguished by indices which reflect the speaker-hearer-attitude combinations involved, as follows:

- (131)
$F_{hs}(P)$	=	the extension H knows S knows P to have
$F_{hsm}(P)$	=	the extension H knows S wants to make known
P to have
$F_{hMKswhs}(P)$	=	the extension H knows H and S mutually know
that S suspects ('w': 'weakly believes') that
H knows that S knows P to have
$F_{hMKswhsm}(P)$	=	the extension H knows H and S mutually know
that S suspects that
H knows S wants to make P known to have

It becomes quite cumbersome to represent these more complex states and state changes in terms of n-tuples of functions, many of which will be unaffected by the changes. Instead, we will describe changes in the format (132):

- (132) $\{ \langle M, M' \rangle \mid M'[\dots]M \ \& \dots \}$

where $M'[\dots]M$ means that M and M' are identical except for the M -components mentioned in ' $[\dots]$ '; the way these components are different in M' is stipulated in the clause ' $\& \dots$ '. Instead of (129) we then obtain (133), which is more complicated than (129) not only because we acknowledge the more complex effects (130), but also because the constant-interpretation functions mentioned in (131) have more than one negative counterpart, corresponding with different scopes of negations. For instance, F_{hsm} has the negative counterparts $F_{h'sm}$ and $F_{hsm'}$; $F'_{hsm}(P)$ describes the extension of P as far as H knows that S does not want it to make known, and $F_{hsm'}(P)$ represents those individuals that H knows S wants to make known not to belong to the extension of P .

$$\begin{aligned}
(133) \quad & || < inform, SLEEP(adam), S, H > || \\
& \{ < M, M' > \mid M'[F_{hsm}, F_{hs}, F_{hMKswasm}, F_{hMKswhs}, \text{ and their} \\
& \text{negative counterparts}]M \ \& \\
& G_{hsm}[SLEEP(adam)]F_{hsm} \ \& G(adam) \in G_{hsm}(SLEEP) \ \& \\
& G_{hs}[SLEEP(adam)]F_{hs} \ \& G(adam) \in G_{hs}(SLEEP) \ \& \\
& G_{hMKswasm}[SLEEP(adam)]F_{hMKswasm} \ \& \\
& G(adam) \in G_{hMKswasm}(SLEEP) \ \& \\
& G_{hMKswhs}[SLEEP(adam)]F_{hMKswhs} \ \& G(adam) \in G_{hMKswhs}(SLEEP) \ \& \\
& G_{h'sm}[SLEEP(adam)]F_{h'sm} \ \& G(adam) \in G_{h'sm}(SLEEP) \ \& \\
& G_{h'sm'}[SLEEP(adam)]F_{h'sm'} \ \& G(adam) \in G_{h'sm'}(SLEEP) \ \& \\
& G_{h's}[SLEEP(adam)]F_{h's} \ \& G(adam) \in G_{h's}(SLEEP) \ \& \\
& G_{h's'}[SLEEP(adam)]F_{h's'} \ \& G(adam) \in G_{h's'}(SLEEP) \ \& \\
& \text{negative counterparts of } G_{hMKswasm}, G_{sMKswhs} \}
\end{aligned}$$

An interesting question is how this analysis of utterance meaning compares to the one that is implicit in the pragmatic theory outlined in the previous section. There, we analysed communicative functions as mathematical functions, applicable to a semantic content plus a context, consisting of the knowledge states of speaker and addressee, respectively. The meaning of an utterance is then obtained by applying its communicative function CF to its semantic content p , while maintaining the abstraction over the context $< K_S, K_H >$:

$$\begin{aligned}
(134) \quad & || < CF, p, S, H > || = CF(p) = \\
& (\lambda K_S, K_H : < updt(K_S, \{z \mid z = KMK(S, H, S \text{ suspects that} \\
& \quad H \text{ knows that } C_j(p), \text{ for some appropriateness} \\
& \quad \text{condition } C_j(p)\}), \\
& \quad updt(K_H, \{z \mid z = H \text{ knows that } C_j(p), \text{ or} \\
& \quad \quad z = KMK(H, S, H \text{ knows that } S \text{ suspects that} \\
& \quad \quad H \text{ knows that } C_j(p), \text{ for some appropriateness} \\
& \quad \quad \text{condition } C_j(p)\}) >)
\end{aligned}$$

Applying this to the example of the informing utterance 'Adam sleeps', (133) and (134) are almost the same, if we disregard the effects on the speaker in (134), even though they may seem quite different at first sight. The *updt* function in (134) changes the hearer's state K_H in such a way that it becomes the case that (1) H knows that $C_j(p)$ for every appropriateness condition for informs, where p

= SLEEP(adam), and (2) H knows that the two participants mutually know that the speaker suspects this to be the case. Representing the knowledge state of H as a state of the kind considered in this section, the *updt* function makes precisely the changes described in (133).

Two differences remain, between (133) and (134). First, (134) takes also changes in the speaker's state into account. Second, (133) is of the general form of a set of pairs (135a), and (134) of the general form of a function (135b):

$$(135a) \quad ||u|| = \{ \langle M_H, M'_H \rangle \mid \dots \}$$

$$(135b) \quad ||u|| = (\lambda M_S, M_H : \langle updt(M_S, \dots), updt(M_H, \dots) \rangle$$

where M_S and M_H represent the states of S and H, respectively.

We think the form (135b) is to be preferred, if the meaning of an utterance is viewed as the way it changes the state(s) of the hearer (and speaker). For the function in (135b) indeed tells us exactly how this (these) state(s) change, whereas the set of changes (135a) only indicates a set of possible ways of changing state(s). This point is closely related to the closing remark in the previous subsection, where we mentioned as a shortcoming of the G & S-approach that the semantic interpretation of a sentence like 'A man sleeps' is the set of all those changes which result either in a state where Adam sleeps, or one where John sleeps, or ...; in short, in a state where a specific man sleeps. Instead of considering all those states, it seems preferable to consider only the change which results in the state where it is known that some man sleeps, without having to choose a particular man.

In fact, we think a state-change approach to meaning should be required to use such models that the state-changing effect of an utterance can always be a single state transition, rather than a set of possible transitions, so that the meaning of an utterance is indeed described as the way it changes an interpreter's (and a speaker's) state of information.

7 Conclusion

The on-line interpretation of natural-language utterances in a (human-computer) dialogue meets with many obstacles. In this paper we have looked at obstacles of a syntactic, of a semantic, and of a pragmatic nature. In each case we have discussed some recently developed tools that seem useful for removing those obstacles, and we have paid separate attention to the possible combination of tools.

As an obstacle of a syntactic nature we have considered the phenomenon of discontinuous constituents. We have defined treelike structures with discontinuities ('discontinuous trees') which allow us to represent the constituent structure of sentences with discontinuous constituents without using transformations or artifacts like slash categories. On the basis of the notion 'adjacency sequence', which generalizes the notion of a sequence of adjacency pairs, we have developed a kind of phrase-structure grammar, called DPSG ('Discontinuous Phrase-Structure Grammar') which is able to generate discontinuous trees.

Discontinuous trees are a generalization of ordinary trees, and DPSG rules are generalizations of ordinary rewrite rules. Both concepts can therefore be added to an ordinary (generalized) phrase-structure grammar, with the effect that such a grammar generates syntactic representations with discontinuities for sentences with discontinuous constituents, while doing everything else as before.

DPSG rules can also be handled by a parser for ordinary phrase-structure grammars, such as a chart parser, by adding some bookkeeping to make sure that no rules are overlooked and that the adjacency sequence relations are satisfied. Such a parser handles sentences with discontinuous constituents in the same on-line fashion as other sentences.

As obstacles of a semantic nature we have looked at (lexical) vagueness and ambiguity. We have outlined a two-level model-theoretic semantic approach which uses intermediate formal representations that preserve the vagueness and (within-category) lexical ambiguity of natural language. This allows us to postpone the resolution of such ambiguity and vagueness, which in general cannot be resolved in a domain-of-discourse-independent way, until we have the information that is required for this purpose. We indicated that some cases of structural, 'logical' ambiguity can also be handled in that way. This approach can be combined with DPSG (or other syntactic formalisms) in a syntactic-semantic grammar where the semantic components of the rules translate syntactic constituents into these intermediate semantic representations. A parser-interpreter which applies the rules of such a grammar can deal with virtually all kinds of ambiguity and vagueness in an on-line manner.

Another obstacle of a semantic nature is formed by anaphora. We have described a suggestion by Groenendijk and Stokhof to handle anaphora using predicate-logic representations that are interpreted as in dynamic logic. In that approach,

the meaning of a representation is conceived as the way a model of the domain of discourse would have to change in order to make the representation true. This leads to a particularly elegant treatment of anaphora by means of free variables. However, the approach has serious deficiencies for its application to natural language. We have indicated some of these deficiencies, and directions for removing them.

As an obstacle of a pragmatic nature, we have considered the interpretation of the communicative function of an utterance in an information-exchange dialogue. We argued that utterance meaning is appropriately defined in terms of the way the interpretation of the utterance changes the states of information of the speaker and addressee, and that this can be viewed as an extension of the dynamic logic idea of Groenendijk and Stokhof. We indicated how a pragmatic theory of utterance function and the dynamic logic approach may be combined into a unified theory of utterance interpretation, which we tentatively called 'dynamic interpretation theory'.

Groenendijk and Stokhof's approach gives rise to an on-line interpretation process for sentences in a text. This can be seen as an instance of the more general phenomenon that the use of semantic representations whose interpretation is defined in terms of state changes, is a suitable basis for on-line interpretation. 'Dynamic interpretation theory' is another instance, where instead of predicate-logic representations of sentences we use meaning representations of utterances, which include, besides the semantic representation, the indication of communicative function, speaker and hearer. These more complex representations are interpreted dynamically in terms of changes in the states of speaker and hearer. Although many details still have to be worked out, this approach offers the exciting prospect of a full-fledged, on-line semantic and pragmatic interpretation of utterances in dialogue.

References

- Allwood, J. (1976) *Linguistic communication as action and cooperation*. Gothenburg Monographs in Linguistics 1.
- Beun, R.-J. (1986) The function of repetitions in spoken information dialogues.

IPO Annual Progress Report 20, 91-98.

Beun, R.-J. (1988) Declarative question acts. In M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogues*. North-Holland, Amsterdam.

Beun, R.-J. & Bunt, H.C. (1988) Investigating linguistic behaviour in human-computer dialogues. IPO Annual Progress Report 22,86 - 96..

Bronnenberg, W.J. Bunt, H.C., Landsbergen, S.P.J., Scha, R.J.H., Schoenmakers, W.J. & Utteren, E.P.C. van (1980) The question- answering system PHLIQA1. In L. Bolc (ed.) *Natural language communication with computers*. Hauser, München.

Bunt, H.C. (1977) Towards an analysis of dialogue organization principles. IPO Annual Progress Report 12, 105-114.

Bunt, H.C. (1984) The resolution of quantificational ambiguity in the TENDUM system. COLING'84 preprints, 130 - 133.

Bunt, H.C. (1985) *Mass terms and model-theoretic semantics*. Cambridge University Press, Cambridge, England.

Bunt, H.C. (1988a) Information dialogues as communicative action in relation to partner modeling and information processing. In M.M. Taylor, F. Néel & D.G. Bouwhuis (eds.) *The Structure of Multimodal Dialogues*. North-Holland, Amsterdam.

Bunt, H.C. (1988b) Information dialogues and the dynamics of language understanding. In H. Bouma & B. Elsendoorn (ed.) *Working models of human perception*. North-Holland, Amsterdam.

Bunt, H.C. (forthc.) A multiple-data base knowledge representation formalism. Unpublished paper.

Bunt, H.C., Beun, R.J., Dols, F.J.H., Linden, J.A. van der, & Schwartzenberg, G.O. thoe (1985) The TENDUM dialogue system and its theoretical basis. IPO

Annual Progress Report 19, 105- 113

Clark, H.H. & Marshall, C.R. (1981) Definite reference and mutual knowledge. In: A.K. Joshi, I. Sag & B.L. Webber (eds.) *Elements of discourse understanding*. Cambridge University Press, Cambridge.

Cohen, P.R. & Levesque, H.J. (1985) Speech Acts and Rationality. Proceedings of the 23rd Annual Meeting of the Assoc. for Computational Linguistics, 49-59.

Emonds, J.E. (1976) *A transformational approach to English syntax*. Academic Press, New York.

Emonds, J.E. (1979) Appositive relatives have no properties. *Linguistics Inquiry* 10, 211-243.

Gazdar, G. (1979a) *Pragmatics: Implicature, Presupposition, and Logical Form*. Academic Press, London & New York.

Gazdar, G. (1979b) Speech act assignment. In: A.K. Joshi, I. Sag & B.L. Webber (eds.) *Elements of discourse understanding*. Cambridge University Press, Cambridge.

Gazdar, G., Klein. E., Pullum, G.K. & Sag, I.A. (1985) *Generalized Phrase-Structure Grammar*. Harvard University Press, Cambridge, MA.

Groenendijk, J. & Stokhof, M. (1987) Dynamic Predicate Logic. Unpublished paper, ITLI, Amsterdam, March 1987

Halpern, J.Y. & Moses, Y.(1984) Knowledge and common knowledge in a distributed environment. In: Proceedings of the Third ACM Conference on Principles of Distributed Computing, 50-61.

Harel, David (1984) Dynamic Logic. In D. Gabbay & F. Guenther (eds.) *Handbook of Philosophical Logic, vol II*. Reidel, Dordrecht 1984

Joshi, A.K. (1982) Mutual Beliefs in Question-Answer Systems. In: N.V. Smith

(ed.) *Mutual Knowledge*, 181-197. Academic Press, London & New York.

Kamp, J.A.W. (1981) A theory of truth and semantic representation. In: J. Groenendijk & M. Stokhof (eds.) *Formal methods in the study of language*. Mathematisch Centrum, Amsterdam.

Levinson, S. (1983) *Pragmatics*. Cambridge University Press.

Lewis, D. (1969) *Convention*. Harvard University Press, Cambridge, MA.

McCawley, J.D. (1982) Parentheticals and Discontinuous Constituent Structure. *Linguistic Inquiry* 13 (1), 91-106

Perrault, C.R. & Cohen, P.R. (1981) Inaccurate reference. In: A.K. Joshi, I. Sag & B.L. Webber (eds.) *Elements of discourse understanding*. Cambridge University Press, Cambridge.

Raphael, B. (1965) A computer program which 'understands'. In AFIPS Conference Proceedings 26, 577-589.

Ross, J.R. (1973) Slifting. In M. Gross, M. Halle & M.P. Schützenberger (eds.) *The formal analysis of natural language*. Mouton, The Hague.

Searle, J.R. (1969) *Speech Acts*. Cambridge University Press.

Thurmair, G. (1988) Semantic processing in speech understanding. In: H. Niemann, M. Lang & G. Sagerer (eds.) *Recent advances in speech understanding and dialogue systems*. Springer, Berlin.

Wall, R.E. (1972) *Introduction to Mathematical Linguistics*. Prentice-Hall, Englewood Cliffs.

Winograd, T. (1983) *Language as a cognitive process*. Addison-Wesley, Reading, MA.

Bibliotheek K. U. Brabant

17 000 01113204 1