UNIVERSITAS SARAVIENSIS

Universität des
Saarlandes

VERBMOBIL
Semantikkonstruktion

Verbmobil Report 6
Januar 1994

The Verbmobil Semantic Formalism (Version 1.3)

The Verbmobil Semantic Formalism (Version 1.3)

Johan Bos Elsbeth Mastenbroek Scott McGlashan Sebastian Millies Manfred Pinkal

Computerlinguistik
Universität des Saarlandes
D-66041 Saarbrücken

Internet: `verbmobil@coli.uni-sb.de`

Abstract

This report describes the semantic formalism developed at Saarbrücken University as part of the Verbmobil project. The formalism is based upon DRT with additional functionality to meet the requirements on semantic construction arising from spoken dialogue translation. We define the syntax of the formalism and illustrate the semantic composition process in detail.

Contents

1 Introduction

The purpose of this report is twofold. First, it gives an account of the current state of the Verbmobil semantic formalism so that other Verbmobil partners can use it as a reference manual. Second, it presents our view on the general role of a semantic formalism within speech-based dialogue translation and thus indicates the direction for further development of the formalism.

In Section 2, we set out the general principles underlying the design of the semantic formalism, and the specific considerations that have lead to its current version. Section 3 describes the structure of the semantic representations and comments on their motivation and consequences for different aspects of semantic processing. In Section 4, we introduce the basic operations which the formalism provides for semantic construction, and give examples to illustrate how these operations work. Section 5 gives an overview of intended extensions of the formalism: we address immediately relevant tasks as well as more general mid-term perspectives.

2 Requirements on a Semantic Formalism

In this section, we present the general methodological principles which underlie the design of the semantic formalism, as well as the requirements on the formalism imposed by speech-based dialogue translation.

2.1 Methodological Principles

We start by enumerating some important general principles which any semantic formalism designed for purposes of practical application should meet.

Implementation Independence The definition of the formalism must include both a description of the data structures and a set of operations which are defined on them. The user need not care how the data structures and operations are implemented below the given level of presentation. This requirement is best met by providing a declarative level of description. The definitions should be enforced in such a way that a user of the formalism will not be allowed to specify ill-formed structures or use undefined operations.

Modularity Semantic representations should be self-contained, i.e. their well-formedness must be determined without regard to information external to the representation. There must be explicit interfaces relating them to other kinds of representations. Ideally, export and import of information should be possible only through declared interface operations.

Theory Independence The semantic formalism should single out useful aspects from different semantic frameworks, and adopt just these aspects as tools for a specific purpose in a consistent way, avoiding commitment to the philosophical backgrounds and various implications the respective theories come with.

Generality The semantic formalism should provide flexible representational means for formulating descriptions of natural language meaning. To guarantee compatibility with requirements and results that will result from later work, it should avoid theoretical commitments and premature decisions as much as possible.

Expressive Power The formalism should allow the representation of whatever kind of semantic information dialogue translation requires.

Efficiency The formalism should be designed in a way that it allows all processes which use semantic representations to operate efficiently.

It should be clear that these principles are guidelines rather than absolute criteria for the design of a formalism; for example, the well-known trade-off between expressive power and efficient processing must be made. In the next section, we comment on the extent to which these principles have been, or should be, realized in the semantic formalism and discuss more precise requirements imposed by its role in speech-based dialogue translation.

2.2 Specific requirements

Figure 1 gives an overview of system components (and, correspondingly, sub-projects and research fields of Verbmobil) relevant to the semantic formalism. The upper part of the circle contains those components which make use of the semantic formalism. The lower left quarter of the circle contains those components on which semantic construction relies, and the lower right quarter shows the general background of the overall system relevant to semantic processing.

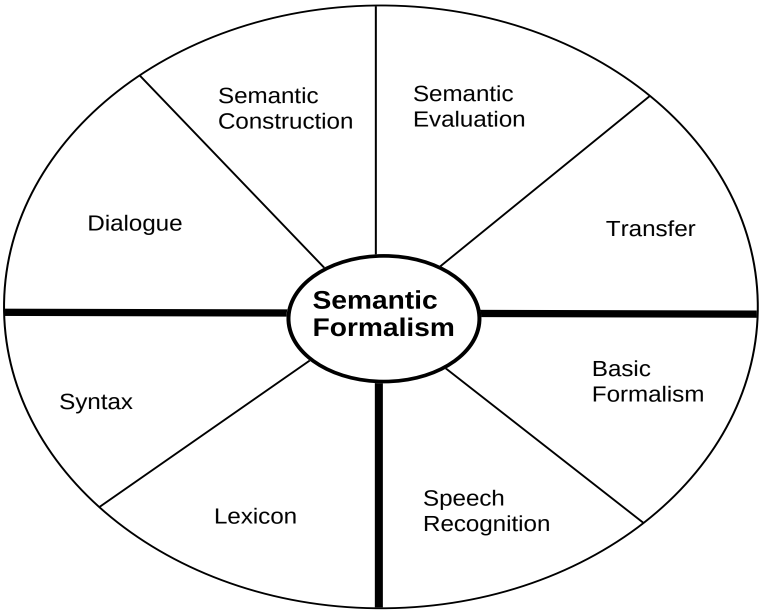


Figure 1: The Semantic Formalism in Verbmobil

Semantic Construction

Semantic construction must be able to deal with local and non-local, sentence-internal as well as intersentential semantic phenomena. Therefore, the semantic formalism must be able to deal with context dependence, especially anaphora in the widest sense. In addition, the semantic formalism must be compositional on the level of semantic representations, in order to meet the conditions of declarativity and modularity (and maybe others). In order to meet the first requirement, we adopt discourse representation structures (DRSs) as basic data structures for the formalism. To fulfil the second requirement, we do away with the obviously non-compositional DRS construction rules of standard DRT (Kamp and Reyle, 1993). Instead, we combine DRSs with the composition method well-known from Montague-style Extended Type Theory (Millies, 1993; Millies and Pinkal, 1993). On this level of presentation, the formalism is entirely declarative and implementation-independent.

Semantic Evaluation

The core function of semantic evaluation is the resolution of ambiguities and referentially underspecified expressions, as required for the task of translation. Semantic evaluation must provide inferencing mechanisms which use information encoded in semantic representations, context information, and extra-linguistic knowledge of different kinds, where the latter is made available by domain modelling.

Accordingly, the semantic formalism must support the task of controlled inferencing, it must keep track of the resolution process, and it must be defined in such a way that it is independent of, but consistent with, the conceptual inventory used in domain modelling. To meet the first requirement, the formalism must have a denotational interpretation. In fact, a large part of our representation language can be given truth-conditionally equivalent first-order predicate logic translations. To support the resolution process, the formalism should allow underspecified and monotonously extendible representations. The current version of the formalism does this only in a rudimentary fashion; for discussions see Section 5.5. The way in which we relate concepts of semantics and of domain modelling is described in Section 3.2.

Transfer

There will be a semantic transfer component, which is intended to be DRS-based. At the time of writing this report, no detailed information about the requirements of the transfer component was available. We are aware that transfer will have additional requirements to those mentioned so far, especially concerning the inclusion of certain surface-oriented structural information in the semantic representation. The formalism in its current version preserves the unit of the word, in that it supports decomposition of word senses only in special cases. Also, a QLF-like approach to underspecification makes phrasal units better visible than they usually are in disambiguated and scoped logical representation (Alshawi, 1992). We are also aware that there might be a need for encoding information about the relationship between a single syntactic structure and the multiple semantic representations which arise from it.

Dialogue

In dialogues, the propositional, truth-conditionally relevant semantic information is usually combined with pragmatic information concerning the performative status, the illocutionary force, or the degree of politeness of the utterance. The dialogue component needs information of the latter kind as one means of identifying the speech act performed with the utterance. We will dedicate a special part of the semantic formalism for encoding pragmatic information; it will not belong to the denotationally controlled part of the formalism, but will be designed according to the needs of the dialogue component. On the other hand, discourse representations must also take into account the pragmatic status of the utterances not only because they are crucial for the way the dialogue memory is built up, but also because they are necessary for establishing anaphoric connections (see Section 5.3).

Further Requirements

Here we briefly mention demands arising from other parts of the system.

- Because signs are used as the basic unit of grammatical description, we need to describe the relation between the semantic representation, one part of a sign, and its other parts. For example, we need to specify the relationship between syntactic and semantic argument requirements.
- In order to interface our semantic formalism with currently available grammar systems, we have to present the formalism in a unification-based environment. We have met this requirement by defining semantic structures in terms of typed feature structures, and defining approximations to the semantic operations in terms of unification. This restriction brings certain limitations; for example, the absence of copying mechanisms leads to well-known difficulties in treating a class of semantic phenomena including ellipsis and co-ordination.
- Speech translation challenges semantics with input of a radically underdetermined and inconsistent kind, both because of the ungrammaticality of spontaneous speech and the unreliability of current speech recognition technology. The requirement to deal with fragmentary input is not taken into account in the current version of the formalism, but is one of the major themes of ongoing work.

3 Syntax of the Formalism

In this section we discuss the syntax of the semantic formalism. We begin by defining the formalism in terms of typed feature structures (Section 3.1). We then discuss some general properties of the formalism (Section 3.2) and subsequently step through each of the definitions addressing more specific issues (Section 3.3). Finally, we describe the relationship between the typed feature structure definitions and the graphical notation (the so-called ‘box’ notation) found in standard DRT (Section 3.4).

3.1 Definitions

The definitions are simply listed in this section; they are explained in the next section. They are given as *typed feature structures* (Pollard and Sag, 1987). The type is indicated by capitals. We use the notation *listof()* to indicate that an instance must be a list of objects of the type given in parentheses. In many cases this is used to represent a set of elements[1].

$$\text{SIGN}\begin{bmatrix} \textit{phon: } \text{PHON} \\ \textit{syn: } \text{SYN} \\ \textit{sem: } \text{SEM} \\ \textit{prag: } \text{PRAG} \\ \textit{dtrs: } \text{DTRS} \end{bmatrix}$$

$$\text{SEM}\begin{bmatrix} \textit{lambda: listof(}\text{LAMBDA\_ELEM}\textit{)} \\ \textit{drs: } \text{DRS} \\ \textit{quants: listof(}\text{SEM}\textit{)} \\ \textit{anchors: listof(}\text{ANCHOR}\textit{)} \end{bmatrix}$$

$\text{LAMBDA\_ELEM} ::= \text{NAMED\_VAR} \mid \text{SEM}$

$$\text{NAMED\_VAR}\begin{bmatrix} \textit{ident: } \text{IDENTIFIER} \\ \textit{sort: } \text{SORT} \end{bmatrix}$$

$\text{SORT} ::= \text{individual} \mid \text{time} \mid \text{state} \mid \text{event} \mid \ldots$

$$\text{ANCHOR}\begin{bmatrix} \textit{param: } \text{NAMED\_VAR} \\ \textit{discourse\_role: } \text{AFS} \\ \textit{constant: @back} \end{bmatrix}$$

$\text{DISCOURSE\_ROLE} ::= \text{speaker} \mid \text{hearer} \mid \ldots$

$$\text{DRS}\begin{bmatrix} \textit{dom: listof(}\text{MARKER}\textit{)} \\ \textit{conds: listof(}\text{CONDITION}\textit{)} \end{bmatrix}$$

$$\text{MARKER}\begin{bmatrix} \textit{ref: } \text{NAMED\_VAR} \\ \textit{ref\_concept: @back} \\ \textit{ref\_agr: } \text{SYNSEM\_AGR} \end{bmatrix}$$

$\text{CONDITION} ::= \text{BASIC\_COND} \mid \text{COMPLEX\_COND}$

$$\text{BASIC\_COND}\begin{bmatrix} \textit{pred: } \text{PRED\_NAME} \\ \textit{pred\_concept: @back} \\ \textit{inst: } \text{NAMED\_VAR} \\ \textit{args: listof(}\text{ARG}\textit{)} \end{bmatrix}$$

$\text{PRED\_NAME} ::= \text{anbieten} \mid \text{dienstag} \mid \ldots$

$$\text{ARG}\begin{bmatrix} \textit{role: } \text{THEM\_ROLE} \\ \textit{role\_concept: @back} \\ \textit{arg: } \text{NAMED\_VAR} \end{bmatrix}$$

$\text{THEM\_ROLE} ::= \text{agent} \mid \text{patient} \mid \ldots$

$\text{COMPLEX\_COND} ::= \text{BOOL\_EXPR} \mid \text{NEG\_EXPR} \mid \text{MODAL\_EXPR} \mid \text{QUANT\_EXPR} \mid \text{ALFA\_EXPR} \mid \text{SUPP\_EXPR} \mid \text{QUEST\_EXPR}$

[1] The basic formalism does not support sets: sets are represented as lists.

$$\mathrm{BOOL\_EXPR}\begin{bmatrix}\textit{bool\_op:}\ \mathrm{BOOL\_OP}\\ \textit{bool\_arg1:}\ \mathrm{DRS}\\ \textit{bool\_arg2:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{BOOL\_OP} ::= \text{if\_then} \mid \text{or}$$

$$\mathrm{NEG\_EXPR}\begin{bmatrix}\textit{neg\_arg:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{MODAL\_EXPR}\begin{bmatrix}\textit{modal\_op:}\ \mathrm{MODAL\_OP}\\ \textit{modal\_mod:}\ \mathrm{MODAL\_MOD}\\ \textit{modal\_arg:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{MODAL\_OP} ::= \text{poss} \mid \text{nec} \mid \ldots$$

$$\mathrm{MODAL\_MOD} ::= \text{zurnot} \mid \text{prinzipiell} \mid \ldots$$

$$\mathrm{QUANT\_EXPR}\begin{bmatrix}\textit{qtr:}\ \mathrm{QUANTIFIER}\\ \textit{var:}\ \mathrm{NAMED\_VAR}\\ \textit{restr:}\ \mathrm{DRS}\\ \textit{scope:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{QUANTIFIER} ::= \text{every} \mid \text{most} \mid \ldots$$

$$\mathrm{SUPP\_EXPR}\begin{bmatrix}\textit{supp\_state:}\ \mathrm{NAMED\_VAR}\\ \textit{supp\_drs:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{ALFA\_EXPR}\begin{bmatrix}\textit{alfa\_arg:}\ \mathrm{NAMED\_VAR}\\ \textit{alfa\_antec:}\ \mathrm{MARKER}\\ \textit{alfa\_restr:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{QUEST\_EXPR} ::= \mathrm{WHQ\_EXPR} \mid \mathrm{YNQ\_EXPR}.$$

$$\mathrm{WHQ\_EXPR}\begin{bmatrix}\textit{whq\_arg:}\ \mathrm{NAMED\_VAR}\\ \textit{whq\_restr:}\ \mathrm{DRS}\end{bmatrix}$$

$$\mathrm{YNQ\_EXPR}\begin{bmatrix}\textit{ynq\_arg:}\ \mathrm{DRS}\end{bmatrix}$$

3.2 General Comments

The definitions in Section 3.1 characterize our semantic formalism in terms of typed feature structures. As will be obvious to those familiar with Discourse Representation Theory (DRT), many of the semantic definitions are based upon those in standard DRT (Kamp and Reyle, 1993). Others, however, will be less familiar and are motivated by the requirements discussed in Section 2 above. For example, the semantic level is characterized not only in terms of a *drs* component, but also three other components. The *lambda* list has been added to allow bottom-up construction of DRSs from syntactic structure. A 'quantifier storage' mechanism is used to provide an underspecified representation of quantifier scoping. An 'anchors' list is used to represent deictic material. These components — as well as other non-standard definitions — are discussed in more detail in the next section and the operations are discussed in more detail in Section 4.1.

Two other aspects of the formalism deserve general comment: components whose definition must be consistent with definitions elsewhere in the system; and components which are externally defined.

Thematic roles and sorts are instances of the first type. The domain model represents concept knowledge pertaining to the application domain (and possibly the discourse domain). These concepts will be represented in the BACK knowledge representation system (Hoppe et al., 1993). In the BACK system, concepts are organized in terms of a sortal hierarchy and, in the case of complex concepts, a set of roles which describe their relationship to other concepts.

Consequently, the domain model will define a hierarchy of sorts and a set of roles. Of these sorts and roles, a subset will be *linguistically relevant*: i.e. they can serve to constrain linguistic combination — for example, a verb may require a 'theme' argument which is of the sort 'human' — and can serve to group together expressions which exhibit similar linguistic behaviour[2]. In addition, it has been argued that semantic principles need access to thematic role information; for example, pronouns can reference a set of individuals in a discourse where set membership is based on sharing the same semantic role (Asher, 1993: 330–1). As a result, the set of sorts and thematic roles defined in the semantic formalism will be defined in conjunction with those defined in the domain model. It should be noted that thematic roles play no direct role in the semantic composition process. Sorts, on the other hand, do play a role in constraining semantic construction: e.g. construction will be unsuccessful in cases where sorts are incompatible. Finally, both sorts and thematic roles have been used in variants of DRT: sorts are merely an elaboration of the simple sorts 'individual', 'time', etc already used in standard DRT; and Bäuerle has already adopted a role-based approach for verbs (cf. Bäuerle, 1988).

Other components in the semantic formalism are defined by other modules. In particular, conceptual components — *ref_concept*, *pred_concept* and *role_concept* — are components defined by the domain model, not the semantic formalism. As a result, they are not part of the **interpreted** semantic formalism. Likewise, the *constant* component in the *anchor* definition will be externally defined by the dialogue module and interpreted within its discourse model.

3.3 Specific Comments

In this section we discuss the definitions in more detail.

(1) $${}_{\textsc{sign}}\left[\begin{array}{l} \textit{phon: } \textsc{phon} \\ \textit{syn: } \textsc{syn} \\ \textit{sem: } \textsc{sem} \\ \textit{prag: } \textsc{prag} \\ \textit{dtrs: } \textsc{dtrs} \end{array}\right]$$

Many contemporary grammar theories, such as HPSG and UCG, adopt a *sign*-based approach to linguistic information: different levels of linguistic information are encapsulated in a single, structured representation called a *sign* (Pollard and Sag, 1987; Calder et al., 1988; Pollard and Sag, forthcoming). In addition to the levels of syntactic and semantic information, our definition of a sign includes a representation of the syntactic 'daughters' and a 'pragmatic' level where the latter can specify, for example, whether the expression contains a performative verb (e.g. *vorschlagen*) or a discourse cue phrase (e.g. *nein*).

Of primary interest to semantic construction is the definition of the semantic level in the sign:

(2) $${}_{\textsc{sem}}\left[\begin{array}{l} \textit{lambda: listof(}\textsc{lambda\_elem}\textit{)} \\ \textit{drs: } \textsc{drs} \\ \textit{quants: listof(}\textsc{sem}\textit{)} \\ \textit{anchors: listof(}\textsc{anchor}\textit{)} \end{array}\right]$$

The semantic level is characterized in terms of four components: a list of lambda elements, a (partial) discourse representation structure (DRS), a list of quantifiers (for underspecification of quantifier scope) and a list of anchors (for deictic expressions).

(3) $\textsc{lambda\_elem} ::= \textsc{named\_var} \mid \textsc{sem}$

The lambda list allows us to adopt a composition approach to DRS construction based on syntactic structure, whilst retaining flexibility over the precise relationship between syntactic and semantic construction processes. Each element on the list can be seen as expressing a semantic requirement: elements on the list are lambda abstractions over the remaining semantic structure. These semantic requirements *can* be correlated with the subcategorization requirements of the grammar. This approach is more flexible than the conventional approach — where semantic requirements are part of the grammatical subcategorization requirements — since there need not be a one-to-one

[2] For example, the passive active relationship is dependent upon the verb having an 'agent'.

mapping between semantic and syntactic requirements. For example, quantifiers such as *jeder* require two 'arguments' — a restriction and a scope — while there are no corresponding syntactic arguments; expletives, which make no semantic contribution, are subcategorized in some syntactic theories; etc. The role of the lambda list in the construction process is discussed in more detail in Section 4.1 below.

We want to be able to represent semantic objects which take arguments of non-basic type, e.g. generalized quantifiers of type $<< e, t >, << e, t >, t >>$ which expect variables of type ¡e,t¿. The ability to represent raised types like this gives us sufficient independence from the syntactic representation of functor argument relations to provide compositional, intuitively simple and unified treatments of several difficult semantic phenomena. In the framework of a unification-based language we cannot express these types directly. However, we can represent them by allowing that the elements on a lambda-list be either basic or complex semantic objects.

(4) $${}_{\text{NAMED\_VAR}}\left[\begin{array}{l}\textit{ident: }\text{IDENTIFIER}\\ \textit{sort: }\text{SORT}\end{array}\right]$$

A simple semantic object is a named variable composed of an identifier and associated sort information. Variables bear the *ident* feature, which identifies them by a unique constant of the meta-language, because it is often necessary to establish the identity or non-identity of variables, e.g. in anaphora resolution or when looking up the information associated with variables in MARKER structures (discussed below). In addition, the sortal type of each variable is given as the value of an additional feature.

(5) SORT ::= individual | time | state | event | ...

The set of sorts will be defined in cooperation with those defined in the domain model as discussed in Section 3.2 above.

Before discussing the main component of the semantic level, the DRS, we briefly comment on the nature of anchors.

(6) $${}_{\text{ANCHOR}}\left[\begin{array}{l}\textit{param: }\text{NAMED\_VAR}\\ \textit{discourse\_role: }\text{DISCOURSE\_ROLE}\\ \textit{constant: @back}\end{array}\right]$$

The main purpose of an anchor is to link a discourse marker introduced by a deictic expression (indicated by the named_var) to a constant in the discourse situation which performs the discourse role. This motivates the tripartite structure of anchors.

(7) DISCOURSE_ROLE ::= speaker | hearer | ...

The complete set of discourse roles will be defined with reference to those used in the dialogue model. Here we have assumed that the discourse situation will be represented in the BACK knowledge representation system.

(8) $${}_{\text{DRS}}\left[\begin{array}{l}\textit{dom: listof(}\text{MARKER}\textit{)}\\ \textit{conds: listof(}\text{CONDITION}\textit{)}\end{array}\right]$$

The main component in the semantics is a DRS defined as a domain of discourse markers, and a list of conditions on these markers.

(9) $${}_{\text{MARKER}}\left[\begin{array}{l}\textit{ref: }\text{NAMED\_VAR}\\ \textit{ref\_concept: @back}\\ \textit{ref\_agr: }\text{SYNSEM\_AGR}\end{array}\right]$$

A discourse marker is defined as a complex structure: a named variable, a domain concept (again assumed to be a BACK term), and agreement information. The SYNSEM_AGR type is a placeholder for agreement information required for reference resolution. Minimally, this will include syntactic agreement information defined by the grammar. It may be extended to include semantic agreement information, such as natural gender and the individual-collective distinction, if required for reference resolution.

(10) CONDITION ::= BASIC_COND | COMPLEX_COND

Conditions on markers can be either basic or complex.

(11) $$_{\textsc{basic\_cond}}\begin{bmatrix} \textit{pred:}\ \textsc{pred\_name} \\ \textit{pred\_concept: @back} \\ \textit{inst:}\ \textsc{named\_var} \\ \textit{args: listof(}\textsc{arg}\textit{)} \end{bmatrix}$$

(12) PRED_NAME ::= anbieten | dienstag | ...

A basic condition is represented as an n-place predicate, together with a concept in the domain model (see below). Each predicate has a name and a primary argument (its 'instantiation') which, however, need not be realized overtly in natural language. With nouns this argument specifies the discourse marker to which the predicate is applied[3]. With verbs, the argument specifies the discourse marker for the described event and the participants in the event are specified in a separate argument list.

(13) $$_{\textsc{arg}}\begin{bmatrix} \textit{role:}\ \textsc{them\_role} \\ \textit{role\_concept: @back} \\ \textit{arg:}\ \textsc{named\_var} \end{bmatrix}$$

(14) THEM_ROLE ::= agent | patient | ...

Each *arg* in the arguments list is characterized in terms of a thematic role, a concept role defined in the domain model, and a *named_var* indicating the discourse referent which plays the role. The set of thematic roles, like sortal types discussed above, will be defined in conjunction with those used in the domain model. The *role_concept* and *pred_concept*, like the *ref_concept* in (9) above, are defined in the domain model.

Complex conditions contain other DRSs.

(15) $$_{\textsc{bool\_expr}}\begin{bmatrix} \textit{bool\_op:}\ \textsc{bool\_op} \\ \textit{bool\_arg1:}\ \textsc{drs} \\ \textit{bool\_arg2:}\ \textsc{drs} \end{bmatrix}$$

(16) BOOL_OP ::= if_then | or

Disjunctions and conditionals are defined as boolean expressions in the usual way.

(17) $$_{\textsc{neg\_expr}}\begin{bmatrix} \textit{neg\_arg:}\ \textsc{drs} \end{bmatrix}$$

(18) $$_{\textsc{quant\_expr}}\begin{bmatrix} \textit{qtr:}\ \textsc{quantifier} \\ \textit{var:}\ \textsc{named\_var} \\ \textit{restr:}\ \textsc{drs} \\ \textit{scope:}\ \textsc{drs} \end{bmatrix}$$

(19) QUANTIFIER ::= every | most | ...

Likewise, the typed feature structure definitions of negation and quantified expressions are analogous to currently standard definitions. For example, quantifiers are represented as duplex conditions: the *var* indicates the variable quantified over, *restr* indicates the restriction, and *scope* the scope of the quantifier.

[3] Thus the basic condition corresponds to a one-place predicate in Predicate Logic.

(20) $$\left[\begin{array}{l} \textit{modal\_op: } \textsc{modal\_op} \\ \textit{modal\_mod: } \textsc{modal\_mod} \\ \textit{modal\_arg: } \textsc{drs} \end{array}\right]_{\textsc{modal\_expr}}$$

(21) $\textsc{modal\_op} ::= \text{poss} \mid \text{nec} \mid ...$

(22) $\textsc{modal\_mod} ::= \text{zurnot} \mid \text{prinzipiell} \mid \ldots$

This definition of modals is similar to that in DRT with the exception of the *modal_mod* component. This has been introduced in order to characterize utterances where constraints are imposed upon the modality. For example, in *Zur Not geht's am Montag* where *geht's* introduces a possibility which is modified by *zur Not*.

The formalism defines three further complex conditions: support conditions, alfa conditions, and question conditions.

(23) $$\left[\begin{array}{l} \textit{supp\_state: } \textsc{named\_var} \\ \textit{supp\_drs: } \textsc{drs} \end{array}\right]_{\textsc{supp\_expr}}$$

We introduce states as primitive objects in our domain. In addition, we introduce a piece of notation which is taken from situation semantics and episodic logic, without sharing the semantic commitments of these frameworks (Barwise and Perry, 1983; Hwang and Schubert, to appear). We write 's $\models$ K' (the state s 'supports' the DRS K) in order to express that s is characterized by whatever makes K true. The support relation creates an intensional context, where the DRS contributes descriptive constraints only.

(24) $$\left[\begin{array}{l} \textit{alfa\_arg: } \textsc{named\_var} \\ \textit{alfa\_antec: } \textsc{marker} \\ \textit{alfa\_restr: } \textsc{drs} \end{array}\right]_{\textsc{alfa\_expr}}$$

Alfa conditions represent the semantics of a class of anaphoric and deictic expressions and presupposed information, including pronouns, definite descriptions, proper names, and other discourse anaphora. Consequently, alfa conditions function as indicators which call for evaluation. In recent theories of presupposition projection, presupposition is taken to subsume anaphora (Heim, 1983; Zeevat, 1991; Van derSandt, 1992; Kamp, 1993). The anaphoric information in an alfa condition must be linked to previously established discourse markers; if this fails, it is projected (accommodated) at a suitable level of discourse, in which case there is a preference for accommodation as global as possible. This ensures that discourse markers for proper names and deixis will be introduced in the main DRS, and hence are fully accessible. While such a *binding and accommodation* mechanism is not part of the semantic formalism, the formalism establishes the representational means for it[4]. In the definition of alfa conditions the *alfa_arg* indicates a distinguished marker, i.e., the marker that is in essence the representative for anaphoric material. The need to make a distinction between markers arises for example from expressions like *the date of a meeting*, where the discourse marker introduced by *the date* is the distinguished marker, and the information that the antecedent must be a date of a meeting forms the descriptive information that is held in *alfa_restr*. Since the latter is a DRS, it can also contain an alfa condition, which therefore allows embedded anaphoric structures (e.g., *meinem terminkalender*).

(25) $\textsc{quest\_expr} ::= \textsc{whq\_expr} \mid \textsc{ynq\_expr}.$

Questions are represented by two different feature structures, one for wh-question and another for yes/no-questions. Unlike alfa conditions, questions are defined so as to subordinate information within the structure. This is motivated on the basis of accessibility: discourse markers introduced in questions cannot be referred to from outside.

(26) $$\left[\begin{array}{l} \textit{whq\_arg: } \textsc{named\_var} \\ \textit{whq\_restr: } \textsc{drs} \end{array}\right]_{\textsc{whq\_expr}}$$

Wh-questions are defined as a DRS condition with two arguments, a variable and a DRS. This conforms to the traditional lambda notation for wh-questions.

[4] We assume that this mechanism is part of the semantic evaluation process.

(27) $_{\text{YNQ\_EXPR}}[\textit{ynq\_arg:}\ \text{DRS}]$

The representation of yes/no-questions is simply defined as a subordinated DRS.

3.4 Comments on Notation

Our semantic formalism can be represented in two distinct ways. The first uses typed feature structures to represent DRSs, according to the syntax provided in Section 3.1 above. This notation is very close to the actual implementation in the basic formalism.

However, the typed feature language syntax obscurs our essential use of λ-abstraction and conversion. To gain clarity and save space, we also use the familiar box notation. What follows here are notational guidelines for this graphical representation. Most of the notation is very similar to the one in Kamp and Reyle (1993).

A DRS is presented graphically as a box, with the discourse markers on the top, ordered horizontally. Distinguished markers are enclosed within square brackets. λ-abstractions are written in the standard way to the left of the box.

We use the following symbols:

i	to range over discourse markers of sort *individual*
t	for markers of time expressions
e	for event discourse markers
s	for markers of sort *state*
x	for markers of any sort
P, Q	for partial DRSs (i.e. variables over properties)

Thematic roles are indicated by an abbreviation of the role within angle brackets subscripted to the variable it defines the role of.

Anchors are characterized with the structure $< \textit{variable}, \textit{discourse\_role} >$ on the right of the box that belongs to the main DRS. Note that the constant in the *anchor* definition is not represented. In fact, all information which is defined outside the semantic formalism, including *ref_concept*, *pred_concept* and *role_concept*, is not given a graphical representation.

The conditions of a DRS are contained in the box, ordered vertically. Basic conditions are represented as normal first order predicate logic formulae, disjunctions as two boxes separated by $\vee$, implication as two boxes separated by $\Rightarrow$.

Negation is indicated by a $\neg$ followed by a box. Modal expressions are represented by either a $\Diamond$ or a $\Box$, that also include a paraphrase of the modality modifier, followed by a DRS. Quantified expressions are shown as duplex conditions.

Alfa expressions are indicated by *alfa: DRS* where the DRS contains a distinguished discourse marker, and similarly wh-questions are represented as *wh_q: DRS* with a distinguished marker. Yes/no-questions are written as the sequence *yn_q: DRS*.

Finally, the graphical notation of the support condition is a state variable followed by $\models$, followed by a DRS.

In addition, the representations may contain the symbols $\otimes$ and $\odot$, denoting the operations of merging and functional composition respectively, which are explained in Section 4.1.

4 Semantic Construction

In this section we show how the semantic representation of a sentence is constructed from the representation of its constituent parts. We start by defining the semantic operations (Section 4.1) and the input syntactic structure (Section 4.2). In Section 4.3 we provide the semantics of the lexical entries of the sentence *Ich schlage den Dienstag vor* and demonstrate how its constituent parts are combined by the semantic operations so as to yield a DRS for the complete sentence. In Section 4.4 we illustrate other phenomena which the formalism is currently capable of representing.

4.1 Semantic Operations

The main operations required for the semantic construction process are: a 'merge' operation, functional composition, and a 'store' operation (as well as its complementary operation 'retrieve') (Millies, 1993).

The 'merge' operation takes two DRSs and returns a new one. The merging of DRSs consists in taking the union of the sets of discourse markers and the sets of conditions separately. This is illustrated in the following CUF implementation:

```
(28) merge(drs_t, drs_t) -> drs_t.

     merge(dom: Xs & conds: C1,
           dom: Ys & conds: C2) :=
                dom: append(Xs, Ys) &
              conds: append(C1, C2).
```

The operation is indicated by $\otimes$ and corresponds to conjunction in Predicate Logic.

The second operation, functional composition, is the basic operation used to combine semantic representations. It is indicated by '$\odot$'. Here is the explanation of functional composition in terms of the λ-calculus:[5]

(29) Functional Composition $\quad \phi \odot \psi =_D \lambda\vec{\sigma}.\phi(\lambda\nu.\psi(\nu)(\vec{\sigma}))$

We call ϕ the 'functor' and ψ the 'argument' of the operation. Note that this rule is a non-standard version of functional composition. It differs in several properties from the rule $\lambda\nu.\phi(\psi(\nu))$ called 'functional composition' in categorial grammar (Steedman, 1985). First, it always binds the first argument position of the argument, instead of forming a sentence with an abstraction over that position. Second, it has no identity, i.e. there is no ψ with $\phi \odot \psi = \phi$ or $\psi \odot \phi = \phi$ for any ϕ. Finally, it has functional application as a special case, i. e. $\phi \odot \psi = \phi(\psi)$ if $\phi :< \tau, \sigma >, \psi : \tau$ and $\phi \odot \psi$ is defined.

The CUF-implementation of this is shown below:

```
(30) compose(sem_t, sem_t) -> sem_t.

     compose(Fun & lambda: [Arg|R] & drs: P &
                   quants: Q1 & anchors: A1,
             Arg & lambda: [_|S] &
                   quants: Q2 & anchors: A2) :=

             lambda: append(S, R) &
                drs: P &
             quants: append(Q1, Q2) &
            anchors: append(A1, A2).
```

The polymorphism expressed above by the $\vec{\sigma}$ notation is captured on the level of the feature-language by simply pre-pending the tail of the argument's lambda-list ('S') to the resulting expression. The unification of `Arg` into the lambda-feature of `Fun` will at the same time take care of the embedded operation application of ψ to its argument. We use the well-known technique of partially evaluating the expressions in the functor's λ-list with regard to functional application (Pereira and Shieber, 1987). The examples in Section 4.3 below illustrate how this works. In the implementation, no α-conversion takes place – as usual in unification based simulations of the λ-calculus. In addition, we have to pass up both the quantifier-stores and the anchors of functor and argument.

[5] For an n-place constant ϕ, $\vec{\sigma}$ denotes a sequence of terms $\sigma_n \ldots \sigma_1$, such that $\phi(\sigma_n) \ldots (\sigma_1)$ is a proposition.

The third operation, quantifier storage, is used in the manner of Nested Cooper Storage (Keller, 1988) to delay the applications of quantifiers in order to give them potentially wide scope:

```
(31) store(sem_t, sem_t) -> sem_t.

     store(Quant & lambda: [lambda: [X|_]],
           Scope & lambda: [X|Rest] &
                   drs: P &
                   quants: Qs &
                   anchors: A) :=

           lambda: Rest &
              drs: P &
           quants: [Quant|Qs] &
          anchors: A.
```

Quantifiers are stored on the *quants* list. The scopings of the quantifiers on this list are not specified. Putting a new quantifier on the list involves saturating an argument position in the scope with the quantifier's 'referential index', i.e. the variable which the quantifier is eventually going to bind. We do not introduce a special notation for storage, but allow this operation as an alternative to ⊙ whenever the types match. An example will be shown in Section 4.4 below.

4.2 Input Syntactic Structure

The semantic construction process operates on the basis of the syntactic structure of expressions contained in the *dtrs* attribute of the sign. In (32) below, the structure for the sentence *Ich schlage den Dienstag vor*, according to the definition of the syntax-semantics interface, is shown in (32) expressed in the basic formalism. We will not comment on it any further, since it is not used in an essential way below.

```
(32) phrase_t&
     dtrs: (head_filler_struc&
            f_dtr: (word_t&A&phon:[ich])&
            h_dtr: (phrase_t&
                    dtrs: (head_comp_struc&
                           c_dtr:[(phrase_t&B&
                                   dtrs:(head_comp_struc&
                                         c_dtr:[(word_t&C&phon:[dienstag])]&
                                         h_dtr: (word_t&phon:[den]&
                                                 syn:subcat:[C])))]&
                           h_dtr:(word_t&phon:[vorschlage]&
                                  syn: (subcat:[A,B] &
                                        sent_type:inverted)))))
```

4.3 A Worked Example

In order to exemplify the composition process, we give the semantic part of the lexical signs for *dienstag*, *den*, *vorschlage* and *ich*, and show how they combine to produce intermediate representations and, eventually, a semantic representation for the sentence *Ich schlage den Dienstag vor*. In cases where the lexical entry is a head and subcategorizes for its complements, the *subcat* list (which is part of the syntax) is also shown. This makes clear the relationship between *sem* values and *subcat* lists since, in these cases, the semantics of the elements on this list corefer with the functor, or argument, of a semantic operation.

In (33) we show the lexical entry for *Dienstag*[6].

(33) (a) **Dienstag**

(b) $$\lambda t. \boxed{\begin{array}{c} \\ \textit{dienstag(t)} \end{array}}$$

(c) $$\begin{bmatrix} \textit{lambda: } \langle \boxed{3} \rangle \\ \textit{drs: } \begin{bmatrix} \textit{dom: } < > \\ \textit{conds: } \left\langle \begin{bmatrix} \textit{pred: dienstag} \\ \textit{inst: } \boxed{3} \; \begin{bmatrix} \textit{sort: time} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textit{quants: } < > \\ \textit{anchors: } < > \end{bmatrix}$$

The semantics of the definite article *den* in (34b) specifies that the semantics of the common noun expression be established in the *alfa* DRS. We can see from the feature structure in (34c) how this is achieved[7].

(34) (a) **den**

(b) $$\lambda P. \lambda Q. \boxed{\textit{alfa: } \; \boxed{\begin{array}{c} [\ x\ ] \\ \\ \\ \end{array}} \quad \otimes \; P(x)} \quad \otimes \; Q(x)$$

(c) $$\begin{bmatrix} \textit{syn: } \begin{bmatrix} \textit{subcat: } \langle \begin{bmatrix} \textit{sem: } \boxed{5} \end{bmatrix} \rangle \end{bmatrix} \\ \textit{sem: } \begin{bmatrix} \textit{lambda: } \left\langle \begin{bmatrix} \textit{lambda: } \langle \boxed{3} \mid \_ \rangle \\ \textit{drs: } \boxed{2} \end{bmatrix}, \begin{bmatrix} \textit{lambda: } \langle \boxed{3} \mid \_ \rangle \\ \textit{drs: } \boxed{4} \end{bmatrix} \right\rangle \\ \textit{drs: } \begin{bmatrix} \textit{dom: } < > \\ \textit{conds: } \left\langle \begin{bmatrix} \textit{alfa\_arg: } \boxed{3} \\ \textit{alfa\_restr: } \begin{bmatrix} \textit{dom: } \langle \begin{bmatrix} \textit{ref: } \boxed{3} \; \begin{bmatrix} \textit{ident: m3} \end{bmatrix} \end{bmatrix} \rangle \\ \textit{conds: } < > \end{bmatrix} \otimes \boxed{2} \end{bmatrix} \right\rangle \end{bmatrix} \otimes \boxed{4} \\ \textit{quants: } < > \\ \textit{anchors: } < > \end{bmatrix} \odot \boxed{5} \end{bmatrix}$$

The lexical entry for *den* subcategorizes for a common noun. The feature structure representation of (34b) is specified as the functor, the semantics of the common noun ($\boxed{5}$) is specified as the argument of functional composition.

[6] In the following examples, lists are represented as < ... > in the feature structures. Note also that the scope of variables is limited to the (sign) feature structure they occur in.

[7] In the following representation, and other representations of type-raised expressions, the *lambda* list is of the form < $\boxed{\text{n}}$ | _ > indicating that it is seeking an argument which is itself seeking $\boxed{\text{n}}$ and, possibly, other unspecified arguments (as indicated by the '_').

Applying the composition rule will unify $\boxed{5}$ with the first expression on the lambda-list of *den*. This has two effects: First, the variable in the common noun expression is unified with the *ref* part of the discourse marker introduced by the article ($\boxed{3}$). Second, the DRS of the common noun is unified with $\boxed{2}$, which forces by merging that the conditions in this DRS appear inside the *alfa* DRS containing the discourse marker. Again by the definition of functional composition, the resulting *sem* value will contain the DRS of the functor, i.e. the article.

Combining the semantics for these constituents by functional composition accordingly yields the following structure for the semantics of the noun phrase:

(35) (a) **den Dienstag**

(b) $$\lambda Q.\ \boxed{\mathit{alfa}:\ \boxed{\begin{array}{l}[\,t\,]\\ \mathit{dienstag(t)}\end{array}}}\ \otimes Q(t)$$

(c) $$\left[\begin{array}{l} \mathit{lambda}: \left\langle \left[\begin{array}{l}\mathit{lambda}: \langle \boxed{3} \mid \_ \rangle \\ \mathit{drs}: \boxed{4}\end{array}\right] \right\rangle \\ \mathit{drs}: \left[\begin{array}{l} \mathit{dom}: <\ > \\ \mathit{conds}: \left\langle \left[\begin{array}{l} \mathit{alfa\_arg}: \boxed{3} \\ \mathit{alfa\_restr}: \left[\begin{array}{l} \mathit{dom}: \left\langle \left[\mathit{ref}: \boxed{3}\ \ \left[\begin{array}{l}\mathit{ident:\ m3}\\ \mathit{sort:\ time}\end{array}\right]\right]\right\rangle \\ \mathit{conds}: \left\langle \left[\begin{array}{l}\mathit{pred:\ dienstag}\\ \mathit{inst}: \boxed{3}\end{array}\right]\right\rangle \end{array}\right] \end{array}\right] \right\rangle \end{array}\right] \otimes \boxed{4} \\ \mathit{quants}: <\ > \\ \mathit{anchors}: <\ > \end{array}\right]$$

The verb *vorschlage* introduces an event variable and assigns roles to its arguments (indicated by bracketing in this graphical representation as discussed in Section 3.4 above).

(36) (a) **vorschlage**

(b) $$\lambda t.\lambda i.\ \boxed{\begin{array}{l} e \\ \mathit{vorschlagen}(e,\ i_{<ag>},\ t_{<th>}) \end{array}}$$

(c)
$$\begin{bmatrix} \textit{syn:} & \begin{bmatrix} \textit{subcat:} & \left\langle \begin{bmatrix} \textit{sem:} & \boxed{7} \end{bmatrix} , \begin{bmatrix} \textit{sem:} & \boxed{8} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textit{sem:} & \boxed{7} \odot \boxed{8} \odot \begin{bmatrix} \textit{lambda:} & \langle \boxed{3} , \boxed{1} \rangle \\ \textit{drs:} & \begin{bmatrix} \textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{6} & \begin{bmatrix} \textit{ident: m6} \\ \textit{sort: event} \end{bmatrix} \end{bmatrix} \right\rangle \\ \textit{conds:} & \left\langle \begin{bmatrix} \textit{pred: vorschlagen} \\ \textit{inst: } \boxed{6} \\ \textit{args:} \left\langle \begin{bmatrix} \textit{role: agent} \\ \textit{arg: } \boxed{1} \end{bmatrix} , \begin{bmatrix} \textit{role: theme} \\ \textit{arg: } \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \right\rangle \end{bmatrix} \\ \textit{quants:} & < > \\ \textit{anchors:} & < > \end{bmatrix} \end{bmatrix}$$

(35), the result of applying *den* to *dienstag* will now be applied to the verb *vorschlage* yielding (37).

(37) (a) **schlage den Dienstag vor**

(b)
$$\lambda i \; \boxed{\begin{array}{l} e \\ \textit{alfa:} \; \boxed{\begin{array}{l} [\, t \,] \\ \textit{dienstag(t)} \end{array}} \\ \textit{vorschlagen(e, } i_{<ag>}, t_{<th>}) \end{array}}$$

(c)
$$\boxed{7} \odot \begin{bmatrix} \textit{lambda:} & \langle \boxed{1} \rangle \\ \textit{drs:} & \begin{bmatrix} \textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{6} & \begin{bmatrix} \textit{ident: m6} \\ \textit{sort: event} \end{bmatrix} \end{bmatrix} \right\rangle \\ \textit{conds:} & \left\langle \begin{array}{l} \begin{bmatrix} \textit{alfa\_arg:} & \boxed{3} \\ \textit{alfa\_restr:} & \begin{bmatrix} \textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{3} & \begin{bmatrix} \textit{ident: m3} \\ \textit{sort: time} \end{bmatrix} \end{bmatrix} \right\rangle \\ \textit{conds:} & \left\langle \begin{bmatrix} \textit{pred: dienstag} \\ \textit{inst: } \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \textit{pred: vorschlagen} \\ \textit{inst: } \boxed{6} \\ \textit{args:} \left\langle \begin{bmatrix} \textit{role: agent} \\ \textit{arg: } \boxed{1} \end{bmatrix} , \begin{bmatrix} \textit{role: theme} \\ \textit{arg: } \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \end{array} \right\rangle \end{bmatrix} \\ \textit{quants:} & < > \\ \textit{anchors:} & < > \end{bmatrix}$$

The entry (38) introduces a new discourse marker for *speaker* which appears in a *alfa* condition. (It must either be linked to a previously introduced discourse referent for the same speaker, or be accommodated in the main DRS.) Furthermore, adding it to the anchor list relates the referent to the *current speaker* via an external call to the BACK system.

(38) (a) **ich**

(b) $$\lambda P.\ \boxed{\begin{array}{ll} \textit{alfa:} & \boxed{\begin{array}{l} [\ i\ ] \\ \\ \\ \end{array}} \end{array}} \ \otimes\ P(i)\ \ < i, speaker >$$

(c) $$\left[\begin{array}{ll}
\textit{lambda:} & \left\langle \left[\begin{array}{l} \textit{lambda: } \langle \boxed{1} \mid \_ \rangle \\ \textit{drs: } \boxed{2} \end{array}\right] \right\rangle \\
\textit{drs:} & \left[\begin{array}{ll}
\textit{dom: } < > & \\
\textit{conds:} & \left[\begin{array}{ll}
\textit{alfa\_arg: } \boxed{1} & \\
\textit{alfa\_restr:} & \left[\begin{array}{l}
\textit{dom: } \left\langle \left[\textit{ref: } \boxed{1} \ \left[\begin{array}{l} \textit{ident: m1} \\ \textit{sort: individual} \end{array}\right] \right] \right\rangle \\
\textit{conds: } < >
\end{array}\right]
\end{array}\right]
\end{array}\right] \otimes \boxed{2} \\
\textit{quants: } < > & \\
\textit{anchors:} & \left\langle \left[\begin{array}{l} \textit{param: } \boxed{1} \\ \textit{discourse\_role: speaker} \end{array}\right] \right\rangle
\end{array}\right]$$

Finally we combine *ich* (38) with (37), which results in (39).

(39) (a) **Ich schlage den Dienstag vor**

(b) $$\boxed{\begin{array}{ll}
e & \\
alfa: & \boxed{\begin{array}{l} [\ i\ ] \\ \\ \\ \end{array}} \\
alfa: & \boxed{\begin{array}{l} [\ t\ ] \\ \textit{dienstag(t)} \\ \end{array}} \\
\multicolumn{2}{l}{\textit{vorschlagen}(e,\ i_{<ag>},\ t_{<th>})}
\end{array}} \quad < i, speaker >$$

$$
\text{(c)}\quad
\begin{bmatrix}
\mathit{lambda:}\ <\ > \\
\mathit{drs:} & \begin{bmatrix}
\mathit{dom:} & \left\langle \begin{bmatrix} \mathit{ref:}\ \boxed{6} & \begin{bmatrix} \mathit{ident:\ m6} \\ \mathit{sort:\ event} \end{bmatrix} \end{bmatrix} \right\rangle \\
\mathit{conds:} & \left\langle
\begin{matrix}
\begin{bmatrix}
\mathit{alfa\_arg:}\ \boxed{1} \\
\mathit{alfa\_restr:} & \begin{bmatrix}
\mathit{dom:} & \left\langle \begin{bmatrix} \mathit{ref:}\ \boxed{1} & \begin{bmatrix} \mathit{ident:\ m1} \\ \mathit{sort:\ individual} \end{bmatrix} \end{bmatrix} \right\rangle \\
\mathit{conds:}\ <\ >
\end{bmatrix}
\end{bmatrix} \\
\begin{bmatrix}
\mathit{alfa\_arg:}\ \boxed{3} \\
\mathit{alfa\_restr:} & \begin{bmatrix}
\mathit{dom:} & \left\langle \begin{bmatrix} \mathit{ref:}\ \boxed{3} & \begin{bmatrix} \mathit{ident:\ m3} \\ \mathit{sort:\ time} \end{bmatrix} \end{bmatrix} \right\rangle \\
\mathit{conds:} & \left\langle \begin{bmatrix} \mathit{pred:\ dienstag} \\ \mathit{inst:}\ \boxed{3} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \\
\begin{bmatrix}
\mathit{pred:\ vorschlagen} \\
\mathit{inst:}\ \boxed{6} \\
\mathit{args:} & \left\langle \begin{bmatrix} \mathit{role:\ agent} \\ \mathit{arg:}\ \boxed{1} \end{bmatrix} , \begin{bmatrix} \mathit{role:\ theme} \\ \mathit{arg:}\ \boxed{3} \end{bmatrix} \right\rangle
\end{bmatrix}
\end{matrix}
\right\rangle
\end{bmatrix} \\
\mathit{quants:}\ <\ > \\
\mathit{anchors:} & \left\langle \begin{bmatrix} \mathit{param:}\ \boxed{1} \\ \mathit{discourse\_role:\ speaker} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

4.4 Further Examples

In the previous section we have exemplified some of the phenomena which the current version of the semantic formalism is capable of representing. In particular, the sentence *Ich schlage den Dienstag vor*, represented in (39), contains a definite description, a deictic expression, a simple temporal expression and states an event.[8]

The example DRSs presented in this section illustrate other phenomena which our formalism is capable of representing. Examples (40) and (41) show the treatment of quantifier storage.

Here is the semantic representation for *jeden Dienstag*.

[8] We have treated this sentence simply as an assertion, ignoring any other performative impact it might have.

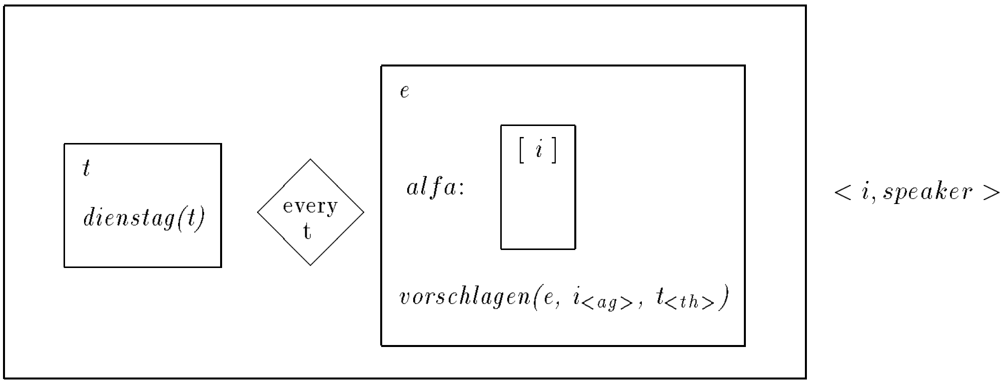
(40)

$$\left[\begin{array}{ll} \mathit{lambda:} & \left\langle \left[\begin{array}{ll} \mathit{lambda:} & \left\langle \boxed{3} \mid \_ \right\rangle \\ \mathit{drs:} & \boxed{8} \end{array}\right] \right\rangle \\ \mathit{drs:} & \left[\begin{array}{ll} \mathit{dom:} & < > \\ \mathit{conds:} & \left\langle \left[\begin{array}{ll} \mathit{qtr:} & \mathit{every} \\ \mathit{var:} & \boxed{3} \\ \mathit{restr:} & \left[\begin{array}{ll} \mathit{dom:} & \left\langle \left[\mathit{ref:}\ \boxed{3} \left[\begin{array}{l} \mathit{ident:}\ m3 \\ \mathit{sort:}\ \mathit{time} \end{array}\right] \right] \right\rangle \\ \mathit{conds:} & \left\langle \left[\begin{array}{ll} \mathit{pred:} & \mathit{dienstag} \\ \mathit{inst:} & \boxed{3} \\ \mathit{args:} & < > \end{array}\right] \right\rangle \end{array}\right] \\ \mathit{scope:} & \boxed{8} \end{array}\right] \right\rangle \end{array}\right] \\ \mathit{quants:} & < > \\ \mathit{anchors:} & < > \end{array}\right]$$

When (40) combines with the verb *vorschlage* (36), the 'store' operation stores the semantics of the noun phrase on the verbs *quants* list. After combining with the subject, the sentence yields the following DRS and feature structure:

(41) (a) **Ich schlage jeden Dienstag vor**

(b)

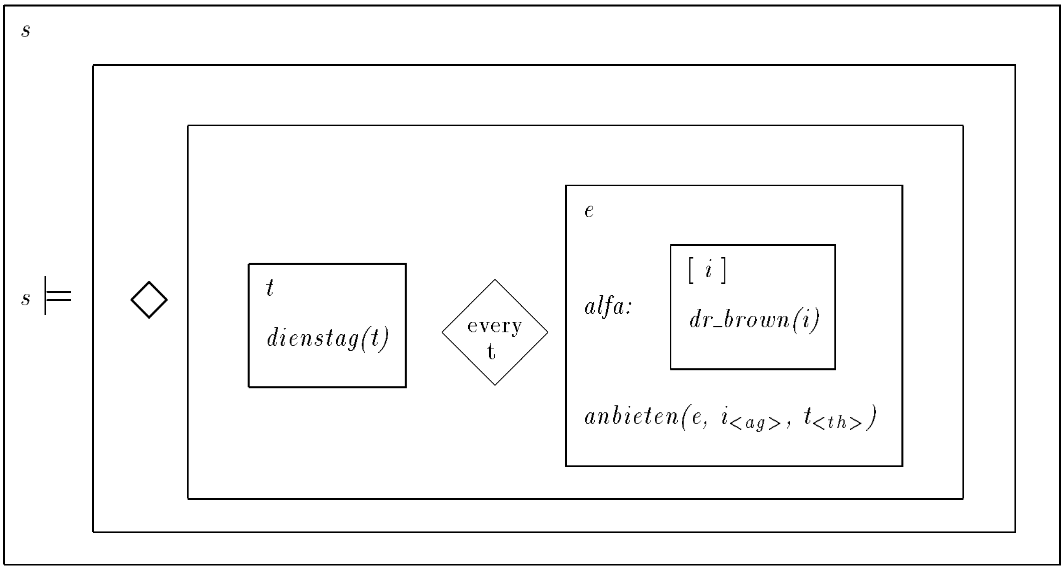


$$
\text{(c)}\quad \begin{bmatrix}
\textit{lambda:} & < > \\
\textit{drs:} & \begin{bmatrix}
\textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{6} & \begin{bmatrix} \textit{ident: m6} \\ \textit{sort: event} \end{bmatrix} \end{bmatrix} \right\rangle \\
\textit{conds:} & \left\langle \begin{matrix} \begin{bmatrix} \textit{alfa\_arg:} & \boxed{1} \\ \textit{alfa\_restr:} & \begin{bmatrix} \textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{1} & \begin{bmatrix} \textit{ident: m1} \\ \textit{sort: individual} \end{bmatrix} \end{bmatrix} \right\rangle \\ \textit{conds:} & < > \end{bmatrix} \end{bmatrix} \\ \begin{bmatrix} \textit{pred:} & \textit{vorschlagen} \\ \textit{inst:} & \boxed{6} \\ \textit{args:} & \left\langle \begin{bmatrix} \textit{role: agent} \\ \textit{arg:}\ \boxed{1} \end{bmatrix}, \begin{bmatrix} \textit{role: theme} \\ \textit{arg:}\ \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \end{matrix} \right\rangle
\end{bmatrix} \\
\textit{quants:} & \left\langle \begin{bmatrix}
\textit{lambda:} & \left\langle \begin{bmatrix} \textit{lambda:} & \langle \boxed{3} \mid \_ \rangle \\ \textit{drs:} & \boxed{8} \end{bmatrix} \right\rangle \\
\textit{drs:} & \begin{bmatrix}
\textit{dom:} & < > \\
\textit{conds:} & \left\langle \begin{bmatrix}
\textit{qtr:} & \textit{every} \\
\textit{var:} & \boxed{3} \\
\textit{restr:} & \begin{bmatrix} \textit{dom:} & \left\langle \begin{bmatrix} \textit{ref:} & \boxed{3} & \begin{bmatrix} \textit{ident: m3} \\ \textit{sort: time} \end{bmatrix} \end{bmatrix} \right\rangle \\ \textit{conds:} & \left\langle \begin{bmatrix} \textit{pred:} & \textit{dienstag} \\ \textit{inst:} & \boxed{3} \\ \textit{args:} & < > \end{bmatrix} \right\rangle \end{bmatrix} \\
\textit{scope:} & \boxed{8}
\end{bmatrix} \right\rangle
\end{bmatrix}
\end{bmatrix} \right\rangle \\
\textit{anchors:} & \left\langle \begin{bmatrix} \textit{param:} & \boxed{1} \\ \textit{discourse\_role:} & \textit{speaker} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Example (42) shows the representations of a proper name, a universal quantifier and a modal. Note that only one reading is presented of the ambiguous sentence (42a).

(42) (a) Dr. Brown kann jeden Dienstag anbieten

(b)

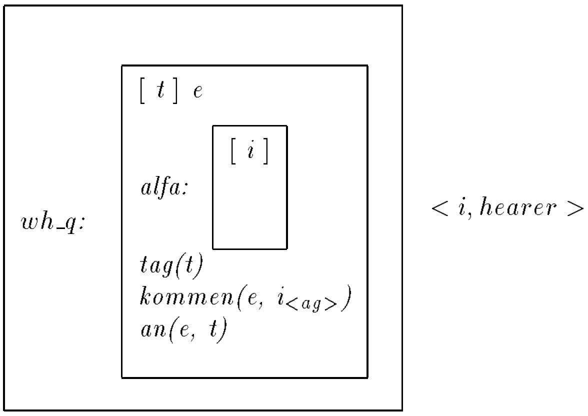


Here we see an example of the intended use of the support-relation: the modal verb takes an event-predicate ('event-type') and introduces a state which is characterized by its being possible that this event-type applies to a certain event.

We conclude this section with the representations of a wh-question and a yes/no-question.

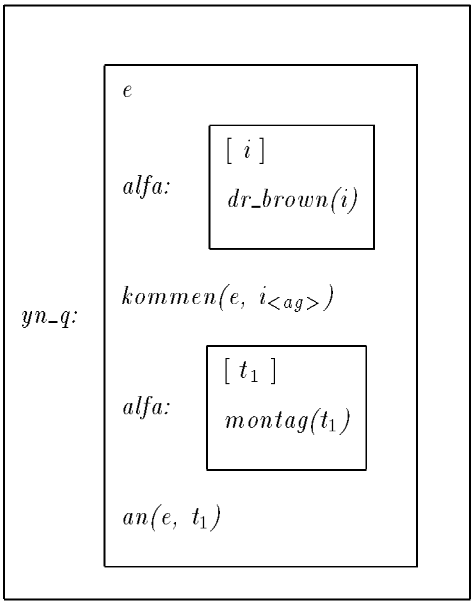
(43) (a) An welchem Tag kommen Sie?

(b)



(44) (a) Kommt Dr. Brown am Montag ?

(b)



5 Further Extensions

As has become clear from several remarks in Section 2, the semantic formalism must be developed further in different respects. First, there are extensions relating to tense, aspect, and plurals, for example, which are less demanding than other aspects of the semantic formalism.

However, there are problems which will require major extensions and, perhaps, revisions of the formalism. The most important ones which we are aware of will be briefly discussed in the following sections.

5.1 Ellipsis and Anaphora

One well known characteristic of spoken dialogues, both human-human and human-computer, is the frequent occurrence of elliptical and fragmentary utterances (Bunt, 1989; MacDermid, 1993). Elliptical utterances share one important feature, namely their anaphoric behaviour. Every elided phrase can be identified by its context: explicitly marked in the previous utterances, or, in a smaller number of cases, implicitly by pragmatic information.

Other phenomena that share characteristics with ellipsis are a class of anaphors that are known to refer to a certain (sub-)structure of discourse. These are for instance *event-type* anaphors, as in the following utterance:

(45) Es geht bei mir am Montag.

where the pronoun *es* refers to parts of the discourse structure introduced by an event description in the discourse. In this case it could refer to type of event ‘having a meeting’.

In an extension of the semantic formalism, ellipsis, and the anaphoric phenomena described above, could be put on a par with other anaphoric expressions on the grounds that they share many characteristics. It is possible then to use alfa conditions to represent elliptical information.

Resolution can be performed in basically two ways. The first way parallels the representation of anaphoric information. It is assumed then that we can refer to pre-established discourse markers which are associated with abstractions to portions of structure in the DRS (Klein, 1987; Asher, 1993). A second possibility for resolution is a copying-and-renaming method, which is found in DRT-implementations (Sem, 1993; Bäuerle, 1988).

5.2 Propositional Objects

Another class of semantic phenomena which has not yet been considered in the formalism is propositional attitudes. We can represent attitude relations by either allowing DRSs as arguments of atomic predicates, or by introducing labels, perhaps also discourse referents, for DRSs into the formalism (Asher, 1993: 111-37). For propositional anaphora it seems that we need these objects anyway. To express ordering relations on quantifier terms in connection with underspecified representations, we also need labels on (partial) DRSs. It seems to be a reasonable modification to introduce names for objects constituting DRSs in general, and to express the internal structure of the DRS by an ordering relation on these labels.

5.3 Semantic-Pragmatic Interaction

Our formalism still needs to address cases where the pragmatic function of an utterance is not explicitly signalled. Identification of pragmatic function is important to the semantic formalism since function can have significant effects upon the semantic representation; effects include constraints upon the accessibility of discourse referents as well as the scope of quantifiers, adverbials and focus-sensitive particles.

In some cases, pragmatic function is linguistically signalled, either in terms of the choice of lexical items or intonation:

(46) I *think* we can meet next Tuesday

(47) I *propose* next Tuesday

(48) you want to meet on Tuesday?

(where '?' indicates question intonation). In such cases, the appropriate semantic representation may be constructed without recourse to pragmatic analysis. With questions, for example, the syntactic part of the sign will provide sentence mood information — `-/+wh`, `-/+inversion` and prosodic information — and this will allow us to represent them as shown in Section 3.1 above. In addition, propositional attitude and performative verbs may be indicated as such in the lexicon — in terms of annotations in their *prag* component — and so allow us to give them a distinctive semantic representation.

However, when pragmatic function is not explicitly marked, the same expression may be assigned different functions according to the discourse context. With questions for example, empirical studies of information dialogues have shown that 20% are expressed as declarative sentences without question intonation (Beun, 1988). Implicit questions may also occur in negotiation dialogues:

(49) (Shall we meet) some afternoon next week.

(50) * Then is fine.

(51) Then would be fine.

Here the status of (49) as a question blocks access to the discourse referent corresponding to *some afternoon next week*: it cannot be referred to by *then* as in (50), but only through modal subordination as in (51)(Roberts, 1987).

In general, when the pragmatic function of an utterance in dialogue is not explicit, building a **single** semantic representation may lead to the definition of inappropriate accessibility relations for discourse referents. Rather: multiple representations should be constructed, reflecting the different pragmatic functions which could be assigned. These representations may interact in a way that can only be described externally by an additional dialogue handling mechanism. Extending our formalism with this 'dialogue-oriented' mechanism will represent a substantial contribution towards the development of a semantic formalism for dialogue (Heisterkamp et al., 1992).

5.4 Syntax-Semantic Interaction

The rules for semantic construction as they now stand are lexicon-centered and entirely non-configurational in nature, i. e. they do not make any reference to properties of the *dtrs* attribute of signs[9]. This works because HPSG reduces all *syntactically* non-local relations to local relations. The treatment of non-local semantic phenomena will to some extent depend on configurational properties of the DRSs. However, at a later stage we expect to take into account also configurational aspects of *syntactic* structure. Preferences for quantifier scope, anaphora resolution, crossover phenomena, information structure and the determination of communicative function may depend on such information.

5.5 Underspecification

The representation of underspecification has so far only been adopted in the formalism at a very basic level: the quantifier storage mechanism (*quants* feature) allows scope-free semantic representations; and the *alfa* expressions can be taken as representations of anaphoric elements with unspecified reference. One straightforward extension of the semantic formalism would be to encode a partial ordering relation on the quantifier store. This would lead to partially specified, monotonically extendible representations of scope relations in the sense of Alshawi, 1992 and Reyle, 1993. Further extensions are required to obtain at least the coverage of the QLF formalism, at least as far as underspecification is concerned: partially specified NP reference, treatment of 'vague relations' as in possession construction, or PPs with nonspecific prepositions (like German *von*) and, as soon as ellipsis is included, also underspecification of elliptical constructions must be modelled. The extension is non-trivial in these cases, however, since currently available techniques must be improved. Yet another important task is the representation of underspecification sourced in syntactic (modifier attachment) ambiguities (Harper, 1992).

All these different extensions of the formalism are very conservative and moderate compared with the changes that are required in order to process material originating from speech input. Here we will frequently face incorrect analyses from the speech component as well as incorrect, incomplete or incoherent utterances leading to incomplete and incoherent syntactic representations (assuming they can be represented at all). Unless we retreat to a completely different alternative way of 'flat semantic analysis' in these cases — which would leave us with the task of unifying its results with the results of the standard analysis — we have to change the formalism so as to allow radically underspecified representations, e.g. of the type that allows some completely unknown relationship to hold between two identified elements, the syntactic relation between which could not be detected.

References

Alshawi, Hiyan (ed.) (1992). *The Core Language Engine.* Cambridge, Mass: MIT Press.

Asher, Nicholas (1993). *Reference to Abstract Objects in Discourse.* Dordrecht: Kluwer Academic Publishers.

Barwise, J. and J. Perry (1983). *Situations and Attitudes.* Cambridge, Mass: MIT Press.

Bäuerle, Rainer (1988). Ereignisse und Repräsentationen. Technical Report LILOG-REPORT 43, IBM Deutschland GmbH.

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

Bunt, Harry C. (1989). Towards a dynamic interopretation theory of utterances in dialogue. In Elsendoorn, Ben A.G. and Herman Bouma (eds) *Working models of human perception.* New York: Academic Press.

Calder, Jo, Ewan Klein, and Henk Zeevat (1988). Unification categorial grammar: a concise extendable grammar for natural language processing. In *COLING 88.* 83–6.

Harper, Mary P. (1992). Ambiguous Noun Phrases in Logical Form. *Computational Linguistics* **18**. 419–65.

[9] except that the *functional* distinction between complements and adjuncts determines the inheritance of semantic values from daughter nodes to their mothers.

Heim, Irene (1983). On the Projection Problem for Presuppositions. In *Proceedings of the West Coast Conference on Formal Linguistics.*

Heisterkamp, Paul, Scott McGlashan, and Nick J. Youd (1992). Dialogue semantics for spoken dialogue systems. In *Proceedings of the International Conference on Spoken Language Processing,* Banff, Canada.

Hoppe, Th., C. Kindermann, J.J. Quantz, A. Schmiedel, and M. Fischer (1993). BACK V5 tutorial and manual. KIT Report 100, Technical University of Berlin.

Hwang, C. H. and L. K. Schubert (to appear). Episodic logic: A situational logic for natural language processing. In *Situation Theory and its Applications*, Volume 3. Standford: CLSI. Proceedings of the Third Conference on Sitation Theory and its Applications (STA-3), Oiso, Kanagawa, Japan, 18th–21st November 1991.

Kamp, Hans (ed.) (1993). *Presupposition.* Amsterdam: University of Amsterdam. Dyana-2, Esprit Basic Research Report 6852.

Kamp, Hans and Uwe Reyle (1993). *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT.* Dordrecht: Kluwer.

Keller, W. (1988). Nested cooper storage. In Reyle, Uwe and Christian Rohrer (eds) *Natural Language Parsing and Linguistic Theories.* Dordrecht: Reidel. 432–447.

Klein, Ewan (1987). VP Ellipsis in DR Theory. *Studies in Discourse Representation Theory and the Theory of Generalised Quantifiers*

MacDermid, Catriona (1993). Features of naive callers' dialogues with a simulated speech understanding and dialogue system. In *Proceedings of EUROSPEECH'93*, Berlin, Germany.

Millies, Sebastian (1993). Compositional interpretation and syntactic information in SCOLD. SAMOS Report 2, University of the Saarland, Germany.

Millies, Sebastian and Manfred Pinkal (1993). Linking one semantic interpretation system to different syntactic formalism. In Pinkal, Manfred, Remko Scha, and Lenhart Schubert (eds) *Semantic Formalisms in Natural Language Processing.* Wadern, Germany: IBFI GmbH. Dagstuhl-Seminar-Report 57.

Pereira, F. and S. Shieber (1987). *Prolog and Natural Language Analysis.* Number 10 in CSLI Lecture Notes. Standford: CSLI.

Pollard, C. and I. A. Sag (1987). *Information-based syntax and semantics.* Number 13 in CSLI Lecture Notes. Standford: CSLI.

Pollard, C. and I. A. Sag (forthcoming). *Head-Driven Phrase Structure Grammar.* Standford: CSLI.

Reyle, Uwe (1993). De aling with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics* **10**. 123–79.

Roberts, Craige (1987). *Modal Subordination, Anaphora and Distributivity.* PhD thesis, University of Massachusetts, Amherst.

Sem, Helle Frisak. (1993). A proposal for ellipsis in DRT. Technical Report 2, Reports in Language, Logic, Information. University of Oslo.

Steedman, M. J. (1985). Combinators and categorial grammar. In Oehrle, R., E. Bach, and D. Wheeler (eds) *Categorial grammars and natural language structures.* Dordrecht: Foris Publications.

Van der Sandt, Rob A. (1992). Presupposition Projection as Anaphora Resolution. *Journal of Semantics* **9**. 333–377.

Zeevat, Hendrik Willem (1991). *Aspects of Discourse Semantics and Unification Grammar.* PhD thesis, University of Amsterdam.