Corega Tabs: Incremental Semantic Composition

Kerstin Bücher, Günther Görz, and Bernd Ludwig
Computer Science Institute and FORWISS
FAU Erlangen, Germany
eMail: {buecher,goerz}@informatik.uni-erlangen.de,
bdludwig@immd5-wv.uni-erlangen.de

Abstract

In parsing natural language, incremental semantics composition is one of the most prominent issues. In the past, numerous approaches have been developed for assigning meaning to noun and verbal phrases and their complements and modifiers. Often, their inferential power is too low for practical applications or the expressiveness of the representation language leads to intractable inference procedures. As an answer to these problems, we discuss an approach that relies on Description Logics for handling this class of semantics construction. We show how a semantic knowledge base can be setup. We exploit the equivalence between Discourse Representation Structures limited to the expressiveness of ALC and ABoxes for validating DRS with respect to a given knowledge base.

1 Generic Dialogue Management in EMBASSI

The long-term goal of our research is to design and implement a generic dialogue system for rational (spoken) dialogues, which helps a user to achieve certain goals in terms of operations of a technical application system – e.g. an information system, a system for controlling devices, or any other kind of problem solving system. Among its design criteria are the ability to recognize users' intentions, to establish corresponding subgoals and control their processing. Furthermore, it shall enable mixed-initiative, flexible and cooperative conversations and provide a high level of robustness as well as scalability in the linguistic and application dimensions, which includes portability to new domains with as little effort as possible. It shall also be possible to integrate linguistic interaction with multimodal forms of input and output, as e.g. with graphical user interfaces and – by means of appropriate devices – the recognition of deictic actions.

To a large extent, our research and development work in the field of dialogue systems is done within the German joint project EMBASSI ("Elektronische Multimediale Bedien- und Service-Assistenz"), sponsored by the German Federal Ministry of Research which aims to provide easy access for everybody to complex technical systems (A/V home theatre, car devices and public terminals), encouraging multi-modal user input. Besides a chunk parser for spoken utterances, our contributions to this project are first, the dialogue manager, second, formal ontologies for several application domains and third, a language generation component to communicate system utterances to the user.

In this paper, we address in detail the issue of semantics construction during parsing natural language input[1]. As will be shown in section 3, the backbone of our incremental approach to composing semantic representations is λ-DRT [Fis96]: The parser builds Discourse Representation Structures (DRS) [KaR93] incrementally, and after each composition step, the satisfiability with respect to a given knowledge base is verified by an ABox consistency test. For this purpose, we exploit the fact that DRSs can be mapped onto ABoxes. In order to carry out all tasks necessary for semantics composition, we need a more general framework however, as there are several issues to consider which cannot be handled by using Description Logics [Don96] only.

2 Overview of the Levels of Utterance Analysis

Except for trivial cases, a direct mapping from a user utterance to a system command cannot be accomplished. In general, we have to take complex speech acts into account, where the interpretation of the utterance's propositional content is determined by its (local) linguistic-pragmatic context in the first place. This, in turn, is to a large extent influenced by (global) discourse-pragmatic features which provide constraints based on the dialogue history and the actual place of the utterance in the dialogue, as, e.g., being the expected answer to a question. Furthermore, the application provides further constraints by restricting the meaning of words and phrases to their particular use within a given thematic framework. Therefore, we have to distinguish several – interleaved – levels in the analysis of user utterances:

- Linguistic analysis on the utterance-local level, which in turn consists of several levels of syntactic and semantic construction (see section 3);
- Semantic evaluation, i.e. evaluation of semantic operators, reference resolution, and additional transformations of the logical form, augmented by specific computations;

[1] This includes some of the open questions mentioned in [Bue01].

- Application-domain specific specialization of the evaluated semantic representation (see[Lud02]);
- Discourse-pragmatic analysis - a proper function of the dialogue manager.

Syntactic Level. The syntactic level involves parsing a lattice of word hypotheses, using a two-step model for syntactic derivation. We describe syntactic analysis in section 3. In a first step, edges in the lattice are grouped in syntactic chunks [Abn91].[2] In a second step, these fragments are combined into bigger units obeying semantic and pragmatic constraints, the applicability of which is checked by valency and case frames of the lexical units.

Currently, we implement chunk grammars in a unification grammar formalism derived from PATR-II [Shi92] and augmented by DRS composition operators. In parsing with unification grammars, constraints are expressed as path equations. Instead of representing feature structures in a separate formalism, they could as well be expressed in DL. As a little experiment with CLASSIC showed, unification can then be achieved by means of the **same-as** construct for attributes, representing coreference. This construct is still a desideratum for grammar development with more powerful DL-systems.

Semantic Level. For semantic representation, we use the framework of Discourse Representation Theory (DRT)[3]. In particular, to provide a strictly compositional construction, our semantic representation formalism is λ-DRT, which combines the substitutional rigidity of λ-calculus with DRT [Kus96]. How the semantic construction is performed incrementally with the syntactic analysis is presented in section 3 Whereas the semantic representations of words are inserted during lexical scanning, their composition is performed by the execution of semantic operators which are attached to the rules of the chunk grammar.

Semantic evaluation. The DRS obtained by the semantic construction step has to be evaluated w.r.t. resolution of references, in particular of anaphors, and DRT-specific operators. For anaphor resolution, we developed a computational framework based on linguistic and pragmatic heuristics in [Fis96][4]. DRSs may contain logical operators, e.g. disjunction and conditional expressions and so called "duplex conditions" representing natural-language quantifiers. Evaluating such DRSs means to apply certain transformations to them. Disjunction will lead to two alternative DRSs. For quantifiers, the scope ambiguity problem can be resolved by applying the "Cooper storage" algorithm to several DRSs representing the different readings. The evaluation of particular natural-language quantifiers as "at-least n" will result in number restrictions. Furthermore, some

[2]This step is performed by a chart parser with a chunk grammar, working primarily with a head-driven bottom-up strategy.

[3]cf. [KaR93] for an introduction to Discourse Representation Theory

[4]For a general theoretical introduction with a similar computational solution, which covers also presupposition resolution, cf. vol. II of [Bla99]

generic computations like temporal and calendrical calculations to determine precise time and date specifications are performed in this step. Currently, all of these computations are implemented in a procedural way. With the availability of new tools such as TRIPLE [Sin02], we see an opportunity to specify these transformations in a uniform and more declarative way which fits very well with our underlying DL representation.

Application-domain specific specialization. In the next step, the evaluated semantic representation is transformed into a domain specific form where the general lexical concepts are replaced by domain concept structures according to the formal ontology of the application.

3 Incremental Semantic Composition

If we want human-computer-dialogues to be natural, we must allow humans to talk to the computer as they do to humans. Spontaneous speech often is incomplete or incorrect, full of interruptions and self-corrections, leading to an ungrammatical input to the parser. Additionally, given the error rates of speech recognizers, even with correct input the speech recognizer may produce an output which is not grammatical. Apart from this, parsing German input is difficult, since German is a language with fairly free word order, also allowing for discontinuous constituents. Therefore, the grammar cannot rely only on linear sequence as its main concept. We try to overcome these problems by designing a two-phase parsing process (as presented in [Bue02]). In this section we describe the two phases of parsing looking at the two levels of syntactic and semantic composition of words to chunks and, hopefully a proposition which – interpreted in its context – results in a system action.

3.1 Chunk Composition

The first phase works with a grammar that employs phrase structure rules to build small phrases, called chunks (similar to [Abn91]). A chunk consists of a head element C_h and not more than one other constituent C_f that is a possible filler of a free position in the head's (X-Bar-) structure.

$C \rightarrow C_1\ C_2$ where one (C_h; $h \in \{1, 2\}$) of the two categories is the head.

The filler usually is a complement (as is a noun phrase NP within a prepositional phrase PP) or a modifier (e.g. an adjective phrase $AdjP$ within an NP)[5]. A

[5]In generative grammar the term *c*omplement is used only for sisters of the lexical head. To avoid confusion we define a new term *c*omplifier that subsumes both complements and adjuncts.

chunk may also consist of only one constituent: $C \to C_1$. If C_1 is the head of the chunk and therefore a terminal lexical category, we get the semantics of C from the lexicon, where the semantic information is stored as a λ-DRS ([Kus96]) Δ. If C_1 is an expanded category[6] it contains the head of the chunk, and the semantics of C is inherited from C_1. So, if there is only one symbol on the right side of the grammar rule, then the *extension* of the left side is determined as follows:

$$\mathrm{ext}(C) := \begin{cases} \Delta & C_1 \text{ is the category and } \Delta \text{ is the DRS of the} \\ & \text{lexicon entry.} \\ \mathrm{ext}(C_1) & \text{otherwise} \end{cases}$$

The *semantic head*[7] of the *chunk* is the one of its DRS:

$$\mathrm{head}(C) := \mathrm{head}(\mathrm{ext}(C))$$

In case of a chunk consisting of a head C_h and another constituent C_f ($h \neq f \in \{1, 2\}$), C_f is related to the discourse referent d of C_h by a role R taken from the inventory of EuroWordNet (see [Lud02]). Syntactically, the combination of two categories to a chunk is determined by a grammar rule, which relates the two constituents via the role R. We then get the *extension* of the chunk by λ-composition of the DRSs of the two constituents (T is a DRS-variable):

$$\begin{aligned} \mathrm{ext}(C) &:= (\lambda T.\Delta_k + T(d_2))\left((\lambda T.\Delta_h + T(d_1))\lambda x.y.\left[\frac{\emptyset}{R(x,y)}\right]\right) \\ &= \Delta_k + \Delta_h + \left[\frac{\emptyset}{R(d_1,d_2)}\right] \end{aligned}$$

So, when combining two elements, the parser checks the compatibility of the morphological features (e.g. agreement in case of the combination of a determiner with an NP) and merges their DRSs resulting in a DRS for the chunk that is consistent with the knowledge base (for details of the consistency check see 3.3). This way, each chunk gets an interpretation already at this early stage. If no further parsing is possible we thereby have means to interpret the whole utterance chunk by chunk.

For example, take the utterance "*Kommt Tatort im ZDF?*" from our Embassi application: To combine the preposition *im* and the NP-chunk *ZDF* which was build using the ($NP \to EN$)-rule we apply the following PP-rule[8]:

[6] An example would be a determiner phrase DP that is build from an NP that in turn is build from the lexical category N.

[7] Note that the syntactic head and the semantic head might not be the same; take the DP "*den Krimi*": the syntactic head of the DP is the determiner "*den*" but the semantic head is the noun "*Krimi*". Both heads of a phrase are defined in the prase structure rules.

[8] The fact that this utterance is a Yes/No-question is irrelevant to phase 1, but word order information (apart from intonation the only indicator of the type of speech act) is stored and made available when the pragmatics of the utterance is computed.

PP: P NP:
head = P:
role = has-value:
P morphfeat position = prepos,
P morphfeat kasrek = NP morphfeat case,
PP vpsynfeat clausetype = NP vpsynfeat clausetype,
PP = P:

$$\lambda P.NP. \left(\delta(P)\left(\delta(NP)\left(\lambda x.y.\left[\frac{\emptyset}{\texttt{has-value}(y,x)}\right]\right)\right)\right)$$

The PP-rule contains syntactic as well as semantic information about the chunk-combination. The DRS for the PP-chunk is achieved by λ-composition of the DRSs of *ZDF* and *im* taken from the lexicon related via the role `has-value`:

$$\left[\frac{i}{\texttt{im-SP}(i)}\right] + \left[\frac{l}{\begin{array}{l}\texttt{TVStation1}(l)\\ \texttt{value}(l, ZDF)\\ \texttt{Name}(ZDF)\end{array}}\right] + \left[\frac{\emptyset}{\texttt{has-value}(i,l)}\right] = \left[\frac{i\,l}{\begin{array}{l}\texttt{TVStation1}(l)\\ \texttt{value}(l, ZDF)\\ \texttt{Name}(ZDF)\\ \texttt{im-SP}(i)\\ \texttt{has-value}(i,l)\end{array}}\right]$$

After applying all phrase structure rules we get three chunks, i.e. the NP *Tatort*, the PP *im ZDF*, and the verb phrase VP *kommt*, that after this first phase have a semantic interpretation on their own. The interpretation of the whole utterance is derived by relating these chunks and their interpretation to each other. This is done in phase two.

3.2 Applying Case Frames to Chunks

The second phase is different from the first phase in that it relates chunks that do not need to be adjacent to each other, so the order of the constituents is not relevant but may be an indicator for preferred readings when disambiguation is called for. Phase 2 relies on a kind of dependency grammar that for each chunk of the first phase gives a list of possible syntactic functions the chunk may have:

$$\begin{array}{rcl} C_1 \text{ has } C_2 & \rightarrow & \langle\texttt{synfunc}\rangle \\ \langle\text{constraint equation}\rangle & & \end{array}$$

e.g.:

$$\begin{array}{rcl} VP \text{ has } PP & \rightarrow & \texttt{adverbial} \\ NP \text{ has } PP & \rightarrow & \texttt{attribute} \\ VP \text{ has } NP & \rightarrow & \texttt{subject} \\ NP \text{ agr case} = \text{nom}, & & \\ NP \text{ agr num} = VP \text{ agr num}. & & \end{array}$$

The options are constrained by the morphological features of the chunk, e.g. an NP-chunk functions as subject only if it has nominative case.
For each chunk there is a case[9] frame for its semantic head that stores information about the valencies[10]. The valencies of each chunk are filled by combining it with other chunks, e.g. building a VP from a verb and an NP that functions as its direct object, or expanding a VP by an adverbial PP. The suitability of the combination of two chunks is determined by the semantic constraints of the application ontology. Take the case frame for *kommen*[11]:

infinitive: `kommen`

syntactic function	thematic role	lexical concept
`subject`	`involved-agent`:	`Program1`
`adverbial`	`involved-location`:	`TVStation1`

From the case frame we derive hypotheses about possible complifiers of a chunk using the syntactic functions. Whether a hypothesis is satisfiable is determined by the concepts of the chunks. If they fit (see 3.3), the DRS can be computed: For a semantic head C_h, its complifier C_k, and a theta role $R = \text{thema}(C_h, \texttt{synfunc})$ that C_k can fill, we get the extension of the modified chunk $\tilde{C_h}$ as follows:

$h := \text{head}(C_h)$, $k := \text{head}(C_k)$

$$\begin{aligned}\text{ext}(\tilde{C_h}) &= (\lambda T.\text{ext}(C_h) + T(\text{h}))(\lambda T.\text{ext}(C_k) + T(\text{k})))\left(\lambda x.y.\left[\frac{\emptyset}{R(x,y)}\right]\right) \\ &= \text{ext}(C_h) + \text{ext}(C_k) + \left[\frac{\text{h k}}{\text{thema}(C_1, \texttt{synfunc})(d_1, d_2)}\right]\end{aligned}$$

In our example, the VP *kommt* can be combined with the adverbial PP *im ZDF* since in the case frame of *kommen* there is a valency for an adverbial with the concept `location`. So we get

$$\left[\frac{i\,l\,k}{\begin{array}{l}\texttt{Run}(k)\ \texttt{TVStation1}(l)\ \texttt{involved-location} \\ \texttt{value}(l, ZDF)\ \texttt{Name}(ZDF)\ \texttt{im-SP}(i)\ \texttt{has-value}(i,l)\end{array}}\right]$$

After λ-composition of the DRS above with the DRS for *Tatort* we have a full DRS for our example utterance that is consistent with our knowledge base.

[9]The term *case* is used in the way of Filmore [Fil69] meaning thematic roles

[10]The term *valency* here is used in a broader sense: it includes not only obligatory elements needed to make a phrase syntactically complete; more than that, the case frames list all semantically and pragmatically suitable modifications and their syntactic representations, e.g. attributes for nouns or adverbials for verbs.

[11]The lexical concept is taken from EuroWordNet [Vos98]

3.3 Consistency Check

A DRS composed according to the algorithm outlined above has to be checked for consistency with respect to the given knowledge base. A DRS which passes the test, is called *admissible*. Given a DRS Δ_1 with discourse referent $d_1 = \text{head}(\Delta_1)$ and DRS Δ_2 with discourse referent $d_2 = \text{head}(\Delta_2)$ related via $R(d_1, d_2)$, we have to verify whether

$$\Delta = \Delta_1 + \Delta_2 + \left[\frac{\emptyset}{R(d_1, d_2)}\right]$$

is admissible. C_1 is the concept d_1 is an instance of, and analogously C_2 for d_2. Formally, Δ is *admissible* if and only if C_2 is a R-filler and C_1 is in the domain of R.

If we map Δ to an ABox A, A is inconsistent if Δ is not admissible. For a concept D with $\forall x : D(x) \leftrightarrow \neg C_2(x)$, we assume

$$C_1 \sqsubseteq \forall R.D,$$

If A was satisfiable, the following would hold:

$$\begin{aligned} d_1 \in C_1 \sqcap \forall R.C_2 \wedge d_1 \in C_1 \sqcap \forall R.D \quad \Leftrightarrow \\ d_1 \in \{x | C_1(x) \wedge \forall y (R(x, y) \rightarrow C_2(y) \wedge D(y))\} \end{aligned}$$

From $R(d_1, d_2)$ it follows that $C_2(d_2) \wedge D(d_2)$ holds in A – in contradiction to $C_2(d_2) \wedge \neg D(d_2)$. For R, two axioms are defined:

$$\begin{aligned} \exists R.\texttt{TOP} &\sqsubseteq D \\ \texttt{TOP} &\sqsubseteq \forall R.W \end{aligned}$$

If R was no restriction for C_1, C_1 is not in the domain of R, i.e. $C_1 \not\sqsubseteq W$. Assuming that Δ is still satisfiable, we get

$$d_1 \in \exists R.C_2 \Rightarrow d_1 \in \exists R.\texttt{TOP} \Rightarrow d_1 \in W$$

– a contradiction to $d_1 \in C_1 \not\sqsubseteq W$. Again, if we try¡ to compute the direct types of d_1, the ABox A is inconsistent.

This DRS is domain independent in that it can be derived without context and without knowledge of the application. but therefore it also fails to connect the utterance to the application specific environment and discourse referents. How this connection is established is presented in [Lud02].

Acknowledgements

We are grateful to Yves Forkl, Martin Klarner and Peter Reiss for many discussions and valuable comments on earlier drafts of this paper.

References

[Abn91] S. Abney, *Parsing By Chunks.* In: R. Berwick, S. Abney, C. Tenny (Eds.), *Principle-based Parsing*. Kluwer, 1991.

[Abn95] S. Abney, *Chunks and Dependencies: Bringing Processing Evidence to Bear on Syntax*. In: *Computational Linguistics and the Foundations of Linguistic Theory*. CSLI-Publications, 1995.

[Bue01] K. Bücher, Y. Forkl, G. Görz, M. Klarner, and B. Ludwig, *Discourse and Application Modeling for Dialogue Systems.* Proc. KI-2001 Workshop on Applications of Description Logics, TU Wien, CEUR Proceedings, Vol. 44, 2001.

[Bue02] K. Bücher, M. Knorr, and B. Ludwig. *Anything to Clarify? Report Your Parsing Ambiguities!.* Proceedings of the 15th European Conference on Artificial Intelligence, July 22-26, 2002. Ed. Frank van Harmelen, Lyon, 2002, p. 465-469.

[Bla99] P. Blackburn and J. Bos, *Representation and Inference for Natural Language.* 2 vols., Saarbrücken, 1999. http://www.comsem.org

[Don96] F.M. Donini, M. Lenzerini, D. Nardi, A. Schaerf, *Reasoning in Description Logics*. In: G. Brewka (editor), *Foundations of Knowledge Representation.* CSLI-Publications, 1996, 191–236.

[Fil69] C. Fillmore. *Universals in Linguistic Theory*, chapter The Case for Case. Holt, Rinehart, and Winston, New York, 1969.

[Fis96] I. Fischer, B. Geistert, and G. Görz, *Incremental Semantics Construction and Anaphora Resolution Using Lambda-DRT.* Proceedings of DAARC-96 (Discourse Anaphora and Anaphor Resolution Colloquium, Ed. S. Botley and J. Glass), July 1996, Lancaster, p. 235-244.

[KaR93] H. Kamp, U. Reyle, *From Discourse to Logic*. Dordrecht: Kluwer, 1993.

[Kus96] S. Kuschert, *Higher Order Dynamics: Relating Operational and Denotational Semantics for λ-DRT*. CLAUS-Report 84, Saarbrücken, 1996.

[Lud02] B. Ludwig, K. Bücher, and G. Görz. *Corega Tabs: Mapping Semanatics onto Pragmatics*, this volume, 2002.

[Shi92] S. M. Shieber, *Constraint-Based Grammaar Formalisms. Parsing and Type Inference for Natural and Computer Languages.* A Bradford Book. Cambridge, Mass.: The MIT Press, 1992.

[Sin02] M. Sintek and S. Decker, *TRIPLE – A Query, Inference, and Transformation Language for the Semantic Web.* International Semantic Web Conference (ISWC), Sardinia, June 2002.

[Vos98] P. Vossen, editor. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Kluwer Academic Publishers, Dordrecht, 1998.