Corega Tabs: Mapping Semantics onto Pragmatics

Bernd Ludwig, Kerstin Bücher, and Günther Görz
Computer Science Institute and FORWISS, Erlangen, Germany
eMail: `bdludwig@immd5-wv.uni-erlangen.de`

Abstract

In this paper, we outline our approach to mapping domain independent semantic representations to extensional statement specific to a given application. Fur this purpose, 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 and explain how task independent semantics representations can be specialized to a domain specific meaning. The paper concludes with some performance results and remarks on desirable features of DL reasoners not implemented up to now.

1 DL Models of Applications

Applications are characterized by a DL terminology which models the concepts used for making propositions about application situations. In [BLG02] it is explained, how domain independent semantic representations for a certain class of natural language phrases can be composed relying on EuroWordNet (EWN) [Vos98] as a linguistic ontology. The remaining question is, how these representations can be mapped onto ABoxes only containing propositions in a application specific ontology (sec. 2). As a prerequisite, we address the issue of how those two ontologies can be linked to form a modular DL knowledge base composed of several smaller parts for covering special purpose ontologies (sec. 4).

Basically, the knowledge base is composed of two parts. The EWN ontology encodes the linguistic meaning of words determined on an empirical basis, whereas the Standard Upper Ontology (SUMO) [NP01] is used as a generic base model for concepts of the application domain. See section 2 for a description of how we use the knowledge base. We present our experiments with Racer in section 3. The modular composition of T-Boxes is topic of section 4, and section 5 gives an overview about further requirements of A-Box reasoning.

Thematic Role	Inverse	Thematic Role	Inverse
has-mero-part	has-holo-part	be-in-state-of	state-of
has-mero-portion	has-holo-portion	in-manner	manner-of
has-mero-location	has-holo-location	involved-source	
has-mero-member	has-holo-member	involved-result	
has-mero-madeof	has-holo-madeof	involved-agent	
involved-location	role-location	involved-target	
is-subevent-of	has-subevent		
causes	is-caused-by		

Table 1: Thematic Roles in EuroWordNet

2 Mapping Semantics onto Pragmatics

This section discusses the mapping from DRS composed during parsing [BLG02] to ABoxes representing propositions on the current application situation. This means ABoxes have to be consistent with respect to a given TBox. The rationale for using DL for constructing natural language semantics is the possibility to eliminate hypotheses constructed by the parser if the corresponding ABoxes are inconsistent. Any other hypothesis which passes this test is called *admissible*.

2.1 EuroWordNet

In order to ease the adaptability of the dialogue system to different domains and to reflect general and domain independent usage of language from that of a specific application the semantics of chunks is expressed in terms of concept expressions taken from the EuroWordNet terminology. EuroWordNet has been developed on the basis of the WordNet semantic net which – in version 1.5 – encodes the meaning of about 80.000 nouns, 60.000 verbs and 16.000 adjectives and adverbs. Beyond being a pure taxonomy of semantic types, EuroWordNet can be used to define complex concepts for complements verbs and nouns may take in the German language. In a DL approach to define them, relations between primitive concepts are expressed by roles whereas several different complements for a lexical base form are stated using conjunction of concepts. The linguistic notion of synonymy can be implemented in a DL knowledge base via concept equivalence, antonymy by the use of the negation operator. Disjunction is the tool to state alternative usage of language – for example of different words for the same semantic notion. An example of such a concept description is given by the following definition:

$$\texttt{GetOn1BeOn1} \sqsubseteq \texttt{Air2} \sqcap \forall \texttt{involved-agent}.\texttt{Program1}$$

2.2 Case Frames

Constraints on complements and modifiers of German words are expressed in terms of case frames which state the valencies of a word and their possible semantic filler types.

$$\exists\texttt{involved-agent.TOP} \sqsubseteq \texttt{Run1}$$
$$\texttt{TOP} \sqsubseteq \forall\texttt{involved-agent.Programme1}$$

The definition encodes the meaning of the German verb *kommen* in the sense of *A film is on in channel TV5.* A GCI axiom is used to define the domain and range of the (thematic) role `involved-agent`. In general, thematic roles are used in a number of case frames, not just in one. This means, more than one GCI has to be included in the linguistic terminology that is used to encode the use of German words that take complments or modifiers. A number of such thematic roles is contained in the EuroWordNet terminology. The whole set of roles is listed in table 1. Thematic roles are defined to be features as the relation between discourse referents is functional.

2.3 Mapping DRSs onto ABoxes

The main question to be discussed is the issue of how to verify the mapping of domain independent – in terms of EuroWordNet to application specific language usage – in terms of a domain model. The general idea is that discourse referents in the domain of discourse refer to instances in the application domain. Such pairs of a discourse referent and a corresponding instance are represented by means of a special role called `has-lex`. In the definition $\texttt{AvEvent} \sqsubseteq \forall\texttt{has-lex.Program1}$, it is claimed that an `AvEvent` is related to a discourse referent of `Program1`. As a consequence, all words to whom `Program1` is assigned as meaning in EWN, designate an instance of `AvEvent`. The DRS

$$\left[\begin{array}{l} \sigma\ d \\ \hline \texttt{AvEvent}(\sigma)\ \texttt{has-lex}(\sigma, d)\ \texttt{Program1}(d) \end{array}\right]$$

can be mapped onto an ABox asserting

$$\texttt{Program1}(d) \wedge \texttt{has-lex}(\sigma, d) \wedge \texttt{AvEvent}(\sigma)$$

The set $\mathcal{R}$ of possible application specific readings of an instance of `Program1` is the set of all concepts (in the application domain) subsumed by the concept $\forall\texttt{has-lex.Program1}$. In general:

$$\mathcal{R}(\mathrm{D} := \{C : C \subseteq \forall\texttt{has-lex}.D\}$$

The second issue concerns to verify relations between individuals in the application domain from (thematic) roles in a DRS claimed by the parser. In a domain independent DRS, discourse referents e_1 and e_K may be related by the thematic roles R_1, ..., R_{K-1}:

$$\left[\begin{array}{l} e_1 \ e_2 \ \ldots \ e_K \\ \hline E_1(e_1) \ R_1(e_1, e_2) \ E_2(e_2) \ \ldots \ R_{K-1}(e_{K-1}, e_K) \ E_K(e_K) \end{array}\right]$$

Additionally, we have (from the domain model)

$$\left[\begin{array}{l} e_1 \ \alpha_1 \ e_K \ \alpha_N \\ \hline E_1(e_1) \ \texttt{has-lex}(e_1, \alpha_1) \ A_1(\sigma_1) \ \texttt{has-lex}(e_K, \sigma_N) \ A_N(\sigma_N) \end{array}\right]$$

The case frames for a particular application allow the parser to draw conclusions on roles that hold between α_1, α_2, ..., α_N as a consequence on the assertions made above on e_1, ..., e_K. With description logics, the consistency of these conclusions may be validated with the help of the following GCI (schema) which is instantiated for each linguistic valency in the EuroWordNet terminology:

$$\begin{array}{r} \exists\texttt{has-lex}.(E_1 \sqcap \exists R_1.(E_2 \sqcap \exists \cdots R_i \cdots .\exists R_{K-1}.(E_K \exists \texttt{has-lex}^{-1}.\texttt{TOP})) \quad \rightarrow \\ A_1 \sqcap \exists S_1.(A_2 \sqcap \exists S_2.(A_3 \sqcap \exists \cdots S_j \cdots .\exists S_{N-1} N.A_N)) \end{array}$$

In this axiom, the R_i are thematic roles, the S_j roles in the application domain. E_i are EuroWordNet concepts, A_j application concepts. The axiom is visualized by the following schema:

$$\begin{array}{rcl} & \texttt{has-lex} & \\ A_1(\alpha_1) & \longrightarrow & E_1(e_1) \\ S_1 \downarrow & & \searrow R_1 \\ A_2(\alpha_2) & & E_2(e_2) \\ R_2 \downarrow & & \searrow S_2 \\ A_{N-1}(\alpha_{N-1}) & & E_{K-1}(e_{K-1}) \\ S_{N-1} \downarrow & & \searrow R_{K-1} \\ A_N(\alpha_N) & \longrightarrow & E_K(d_K) \\ & \texttt{has-lex} & \end{array}$$

In order to explain the "genesis" of the axiom schema above, we have to consider several cases of how thematic roles are mapped onto application specific ones.

2.4 Mapping Thematic Roles onto Pragmatic Ones

The interpretation of thematic roles in terms of roles in the application domain is encoded as the application specific part of the case frames describing the language usage in the application domain as in the following example:

$$\begin{array}{r} \exists\texttt{has-lex}.(\texttt{Program1} \sqcap \\ \exists\texttt{involved-time}.(\texttt{um-tp} \sqcap \exists\texttt{value}.(\texttt{clocktime} \sqcap \exists\texttt{has-lex}^{-1}.\texttt{TOP}))) \quad \rightarrow \\ \exists\texttt{has-starttime}.\texttt{TimeInterval} \end{array}$$

Given two DRSs, with the help of an ABox consistency test, one has to validate the application specific reading constructed by the parser. For the utterance "*Sendung um 20.15*", the parser maps the semantic representation on the left onto its application specific interpretation on the right:

$$\left[\begin{array}{l} d\ t\ c \\ \hline \texttt{Program1}(d)\ \texttt{involved-time}(d,t) \\ \texttt{um-tp}(t)\ \texttt{value}(t,c)\ \texttt{clocktime}(c) \\ \texttt{has-hour}(c,20)\ \texttt{has-minute}(c,15) \end{array}\right] \quad \left[\begin{array}{l} \alpha\ \iota \\ \hline \texttt{AvEvent}(\alpha)\ \texttt{has-lex}(\alpha,d) \\ \texttt{TimeInterval}(\iota)\ \texttt{has-lex}(\iota,c) \\ \texttt{has-starttime}(\alpha,\iota) \\ \texttt{has-hour}(\iota,20)\ \texttt{has-minute}(\iota,15) \end{array}\right]$$

2.5 Instantiating Parameters and is-part-of Relations

Given a thematic role between two discourse referents from two different DRSs where the first one serves as functor and the second one as complifier[1], and the individuals corresponding to the discourse referents, the relation to be validated does not necessarily hold between those two individuals. However, the pragmatic relation may hold between the individual corresponding to the discourse referent selected from the functor and an individual accessible via a role path from the individual corresponding to the discourse referent selected from the complifier's DRS as in the following example (DRS: left, A-Box: right):

$$\left[\begin{array}{l} s\ d \\ \hline \texttt{Recording1}(s)\ \texttt{TVStation1}(d) \\ \texttt{involved-patient}(s,d) \\ \texttt{value}(d,\text{ZDF})\ \texttt{Name}(\text{ZDF}) \end{array}\right] \left[\begin{array}{l} s\ d\ \alpha\ \delta\ \sigma \\ \hline \texttt{Record}(\sigma)\ \texttt{has-lex}(s,\sigma)\ \texttt{has-lex}(d,\delta) \\ \texttt{AvEventLocation}(\delta)\ \texttt{AvEvent}(\alpha) \\ \texttt{has-aveventlocation}(\alpha,\delta)\ \texttt{String}(\text{ZDF}) \\ \texttt{has-value}(\delta,\text{ZDF})\ \texttt{has-avevent}(\sigma,\alpha) \end{array}\right]$$

In this example, the thematic role `involved-patient` is mapped onto the role path `has-avevent`∘`has-aveventlocation` establishing the following the axiom:

$$\begin{aligned} &\exists\texttt{has-lex}.(\texttt{Recording1} \sqcap \exists\texttt{involved-patient}.(\texttt{TVStation1} \sqcap \exists\texttt{has-lex}^{-1}.\texttt{TOP})) \rightarrow \\ &\qquad \texttt{Record} \sqcap \exists\texttt{has-avevent}.(\texttt{AvEvent} \sqcap \exists\texttt{has-aveventlocation}. \\ &\qquad\qquad (\texttt{AvEventLocation} \sqcap \exists\texttt{has-value}.\texttt{String})) \end{aligned}$$

With this GCI, we verify that α is a valid parameter for the action σ and that δ forms a part of the instance of `AvEvent` designated by α.

2.6 Instantiating has-part Relations

Often, it happens that a thematic role between two discourse referents induces more than one relation between individuals in the application domain. For example, in the DRS

[1] *Complifier* means *complement or modifier* as explained in [BLG02].

$$\left[\begin{array}{l} v_1\ v_2\ v_3\ v_4\ v_5\ c_3\ c_4 \\ \hline \texttt{TimeDate}(v_1)\ \texttt{has-month}(v_1, c_3)\ \texttt{Month2CalendarMonth1}(c_3)\ \texttt{month-value}(c_3, c_4) \\ \texttt{Number1}(c_4)\ \texttt{value}(c_4, 1)\ \texttt{on}(1) \\ \texttt{has-date}(v_1, v_2)\ \texttt{lc-date}(v_3)\ \texttt{has-day}(v_3, v_4)\ \texttt{Day4}(v_4)\ \texttt{day-value}(v_4, v_5) \\ \texttt{Number1}(v_5)\ \texttt{value}(v_5, 5)\ \texttt{on}(5) \end{array}\right]$$

ABox Assertion	Large KB	Small KB
(most-specific-instantiators v_4 edge2)	2173	134

Table 2: Performance Evaluation for Racer Version 1.6.3 (in milliseconds)

$$\left[\begin{array}{l} s\ d \\ \hline \texttt{Switch1}(s)\ \texttt{involved-patient}(s, d)\ \texttt{Device1}(d) \end{array}\right],$$

two roles have to be instantiated in order to construct the application specific reading: `has-eib`[2] and `has-state`. This may be validated with an GCI axiom:

$$\exists\texttt{has-lex}.(\texttt{Switch1} \sqcap \exists\texttt{involved-patient}.(\texttt{Device1} \sqcap \exists\texttt{has-lex}^{-1}.\texttt{TOP})) \rightarrow \texttt{EIBSwitch} \sqcap \exists\texttt{has-eib}.\texttt{EIBDevice}$$

$$\exists\texttt{has-lex}.(\texttt{Switch1} \sqcap \exists\texttt{involved-patient}.(\texttt{Device1} \sqcap \exists\texttt{has-lex}^{-1}.\texttt{TOP})) \rightarrow \exists\texttt{has-state}.\texttt{ONState})$$

The corresponding ABox may be represented as follows:

$$\left[\begin{array}{l} s\ d\ \sigma\ \delta \\ \hline \texttt{EIBSwitch}(\sigma)\ \texttt{has-lex}(\sigma, s)\ \texttt{has-eib}(\sigma, \delta) \\ \texttt{EIBDevice}(\delta)\ \texttt{has-lex}(\delta, d) \\ \texttt{has-state}(\sigma, \mathrm{o})\ \texttt{OnState}(\mathrm{o}) \end{array}\right]$$

In such a case `has-state` and `has-eib` are called *has-part* relations, as the thematic role `involved-patient` in the semantic representation of the input utterance leads to the instantiation of the individuals o and δ which are part of the definition of σ.

3 Experiments with RACER

We evaluated the performance of the outlined approach in terms of computation time needed on a Pentium III 800 MHz PC running under SuSE Linux 7.2 with 256 MB of RAM. First, we compiled a big knowledge base consisting of

[2] `EIB` = Electronic Instruments Bus

1165 concept definitions and a large number of additional `disjoint` statements. This knowledge base contains the complete SUMO ontology encoded in DL, the EWN upper ontology and the concept definitions specific to the EMBASSI applications. The performance test was carried out by parsing the prepositional phrase *am ersten Januar*. At a certain step during the parsing the consistency of the following ABox has to be tested and the most specific concepts of the DRS's head have to be computed. We get the result in table 2.

Of course, more than 2 seconds for a single call to `most-specific-instantiators` is not acceptable for parsing natural language, as – given a quite complex word lattice – hundreds of such calls have to be performed for parsing one lattice under the constraint of real time behavior of the overall system.

Obviously, however, many SUMO and EWN concepts could be deleted from the knowledge base as they were not used by the application specific part of the knowledge base. In an automatic precompilation step, we deleted 862 concepts which are only defined, but not used as part of another definition – many among them about insects and bacteria which are not considered relevant for the application. The performance test was then repeated (cf. table 2), taking only 134 ms.

As the table indicates, RACER performed faster with the smaller knowledge base, and performance becomes practical. However, the ratio of deleted concepts would be worse in more complex domains. As we did not include the whole set of EWN concepts (about 100.000 concepts), in a more complex application the EWN portion even of the reduced knowledge base would be larger. So, faster classification is what we are dreaming of ...

4 Modular Composition of T-Boxes

Constructing and maintaining large knowledge bases is a task for which methods borrowed from software engineering are of help. The most important principles are modularization and decomposition. As mentioned already, in addition to the separation between discourse and application, we maintain two upper models for lexical semantics and the design of application specific domain models. The first one is covered by EuroWordNet, the second one by the Standard Upper Ontology. As EuroWordNet only contains concepts for nouns, verbs, adjectives, and adverbs, we have to complement this linguistic domain model by separate models for defining notions for temporal and spatial expressions – to mention only the most prominent ones.

In this process, software tools such as OilEd and Protegé can support to a very large extent the engineering task of building such libraries of domain models. Their visual presentation of concept hierarchies helps avoiding duplicate definitions as well as forgetting crucial *disjoint* statements. The possibility to detect incoherent definitions by combining editors and DL reasoners improves the quality of the resulting domain models and speeds up their development.

5 Further Requirements for A-Box Reasoning

In general, domain-instantiated A-Boxes are partial models for application operations. Its contents can be modified by user or system messages. As a framework for processing partial information, we found out that FIL [Abd95] meets all our requirements. We started with the implementation of a prover for a Horn clause subset of FIL in Prolog technology, which has later been replaced by a tableau-based reasoner, operating as a separate module. Such a separate model could be avoided if hypothetical reasoning could be done with A-Boxes in a direct way: Starting with a "core" A-Box containing "hard", but partial information, alternative extensions with hypotheses have to be computed and checked for consistency and completion. These hypotheses are either already given or have to be asked from the user, which amounts to a mixed-initiative subdialogue. Seen as a tableau, leaves of open branches represent possible information states of the dialogue manager at a given time. Inferences on tableau consistency are drawn using domain concept definitions the constraints of which determine justifications, i.e. possibly underspecified parameters corresponding to optional phrases. In other words, a context mechanism is required which manages alternative contexts consisting of a common core with different extensions depending on particular sets of assumptions. From a technical point of view we have to deal with a belief revision problem which could be handled by a reason maintenance mechanism for A-Boxes (cf. [Neb90, Rei87]).

References

[Abd95] N. Abdallah, *The Logic of Partial Information*. New York: Springer, 1995

[BLG02] K. Bücher, G. Görz, and B. Ludwig, *Corega Tabs: Incremental Semantic Composition*, in this volume

[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

[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

[Neb90] B. Nebel, *Reasoning and Revision in Hybrid Representation Systems*. Berlin: Springer (LNAI 422), 1990

[NP01] I. Niles and A. Pease. Toward a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, 2001.

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

[Rei87] R. Reiter and J. de Kleer, *Foundations of Assumption-Based Truth Maintenance Systems*. Proceedings of AAAI-87, Palo Alto: AAAI, 1987, 183–188