AUTOMATIC CONSTRUCTION OF DISCOURSE REPRESENTATION STRUCTURES

Franz Guenthner
Universität Tübingen
Wilhelmstr. 50
D-7400 Tübingen, FRG

Hubert Lehmann
IBM Deutschland GmbH
Heidelberg Scientific Center
Tiergartenstr. 15
D-6900 Heidelberg, FRG

Abstract

Kamp's Discourse Representation Theory is a major breakthrough regarding the systematic translation of natural language discourse into logical form. We have therefore chosen to marry the User Specialty Languages System, which was originally designed as a natural language frontend to a relational database system, with this new theory. In the paper we try to show taking - for the sake of simplicity - Kamp's fragment of English how this is achieved. The research reported is going on in the context of the project ***Linguistics and Logic Based Legal Expert System*** undertaken jointly by the IBM Heidelberg Scientific Center and the Universität Tübingen.

1 Introduction

In this paper we are concerned with the systematic translation of natural language discourse into Discourse Representation Structures as they are defined in Discourse Representation Theory (DRT) first formulated by Kamp (1981). This theory represents a major breakthrough in that it systematically accounts for the context dependent interpretation of sentences, in particular with regard to anaphoric relations.

From a syntactic point of view, however, Kamp chose a very restricted fragment of English. It is our goal, therefore, to extend the syntactic coverage for DRT by linking it to the grammars described for the User Specialty Languages (USL) system (Lehmann (1978), Ott and Zoeppritz (1979), Lehmann (1980), Sopeña (1982), Zoeppritz (1984)) which are comprehensive enough to deal with realistic discourses. Our main tasks are then to describe

- the syntactic framework chosen
- Discourse Representation Structures (DRSs)
- the translation from parse trees to DRSs

The translation from parse trees to DRSs will, as we shall see, not proceed directly but rather via Intermediate Structures, which were already used in the USL system. Clearly, it is not possible here to describe the complete process in full detail. We will hence limit ourselves here to a presentation Kamp's fragment of English in our framework.

The work reported here forms part of the development of a Natural Language Analyzer that will translate natural language discourse into DRSs and that is evolving out of the USL system. We intend to use this Natural Language Analyzer as a part of a legal expert system the construction of which is the objective of a joint project of the University of Tübingen and the IBM Heidelberg Scientific Center.

2 Syntax

2.1 Syntactic framework and parsing process

The parser used in the Natural Language Analyzer was originally described by Kay (1967) and subsequently implemented in the REL system (Thompson et. al. (1969)). The Natural Language Analyzer uses a modified version of this parser which is due to Bertrand &al (1976, IBM (1981)).

Each grammar rule contains the name of an ***interpretation routine***, and hence each node in the parse tree for a given sentence also contains the name of such a routine. The ***semantic executer*** invokes the interpretation routines in the order in which they appear in the parse tree, starting at the root of the tree.

2.2 Syntactic coverage

The syntactic coverage of the Natural Language Analyzer presently includes

- Nouns
- Verbs
- Adjectives and adjectival phrases: gradation, modification by modal adverbial, modification by ordinal number
- Units of measure
- Noun phrases: definiteness, quantification, interrogative pronouns, personal pronouns, possessive pronouns, relative pronouns
- Verb complements: subjects and nominative complements, direct objects, indirect objects, prepositional objects
- Noun complements: relative clauses, participial attribute phrases, genitive attributes, appositions, prepositional attributes
- Complements of noun and verb: negation, locative adverbials, temporal adverbials
- Coordination for nouns, noun phrases, adjectives, verb complexes and sentences
- Comparative constructions
- Subordinate clauses: conditionals
- Sentences: declarative sentences, questions, commands

2.3 Syntax rules to cover the Kamp fragment

In this section we give the categories and rules used to process the Kamp fragment. The syntax rules given below are somewhat simplified with regard to the full grammars used in the Natural Language Analyzer, but they have been formulated in the same spirit. For a detailed account of the German syntax see Zoeppritz (1984), for the Spanish grammar see Sopeña (1982).

Syntactic categories
We need the following categories: <NAME>, <NOMEN>, <QU>, <NP> (features: REL, PRO, NOM, ACC), <VERB> (features: TYP=NI, TYP=NA), <SENT>, <SC> (feature: REL).

Vocabulary
The vocabulary items we have taken from Kamp (1981).

<NAME>	: Pedro, Chiquita, John, Mary, Bill, ...
<NOMEN:+NOM,+ACC>	: farmer, donkey, widow, man, woman, ...
<VERB:TYP=NI>	: thrives, ...
<VERB:TYP=NA>	: owns, beats, loves, admires, courts, likes, feeds,...
<QU>	: a, an, every
<NP:+PRO,+NOM>	: he, she, it
<NP:+PRO,+ACC>	: him, her, it
<NP:+REL,+NOM>	: who, which, that
<NP:+REL,+ACC>	: whom, which, that

2.3.1 Syntax rules

To help readability, the specification of interpretation routines has been taken out of the left hand side of the syntax rules and has been placed in the succeeding line. The numbers appearing as parameters to interpretation routines refer to the position of the categories on the right hand side of the rules. As can be seen, interpretation routines can be nested where appropriate. The operation of the interpretation routines is explained below.

```
 1. <NP>        <- <NAME>
        PRNAME(1)
 2. <NP>        <- <QU> <NOMEN>
        NPQUAN(1,2)
 3. <NOMEN>     <- <NOMEN> <SC:+REL>
        RELCL(1,2)
 4. <SC:+REL>   <- <NP:+REL> <VERB:TYP=NI>
        NOM(VERB(1),1)
 5. <SC:+REL>   <-                 <NP:+REL,+NOM>
    <VERB:TYP=NA>
                <NP:-REL>
        NOM(ACC(VERB(2),3),1)
 6. <SC:+REL>   <-                 <NP:+REL,+ACC>
    <VERB:TYP=NA>
                <NP:-REL>
        ACC(NOM(VERB(2),1),3)
 7. <SC>        <- <NP> <VERB:TYP=NI>
        NOM(VERB(2),1)
 8. <SC>        <- <NP:+NOM> <VERB:TYP=NA> <NP>
        NOM(ACC(VERB(2),1),3)
 9. <SENT>      <- <SC>
        STMT(1)
10. <SENT>      <- if <SC> then <SC>
        STMT(COND(1,2))
```

3 Intermediate Structures

Intermediate Structures are used to facilitate the translation from parse trees to the semantic representation language. They are trees containing all the information necessary to generate adequate expressions in the semantic representation language for the sentences they represent.

3.1 The definition of Intermediate Structures

The basic notions used in Intermediate Structures are RELATION and ARGUMENT. In order to come to adequate meaning representations it has also to be distinguished whether RELATIONs stand for verbs or nominals, therefore the notions VERBSTR and NOMSTR have been introduced in addition. In case of coordinate structures a branching is needed for the ARGUMENTs. It is provided by COORD. Information not needed to treat the Kamp fragment is left out here to simplify the presentation.

3.1.1 Relation nodes and Argument nodes

Nodes of type Relation contain the relation name and pointers to first and last ARGUMENT.

Nodes of type Argument contain the following information: type, standard role name, pointers to the node representing the contents of the argument, and to the previous and next ARGUMENTs.

3.1.2 Verb nodes

Verb nodes consist of a VERBSTR with a pointer to a RELATION. That is verb nodes are Relation nodes where the relation corresponds to a verb. Verb nodes (VERBSTR) contain a pointer to the RELATION represented by the verb. They can be ARGUMENTs, e.g., when they represent a relative clause (which modifies a noun, i.e. is attached to a RELATION in a nominal node).

3.1.3 Nominal nodes

Nominal nodes are Argument nodes where the ARGUMENT contains a nominal element, i.e. a noun, an adjective, or a noun phrase. They contain the following information in NOMSTR: type on noun, a pointer to contents of NOMSTR, congruence information (number and gender), quantifier, a pointer to referent of demonstrative or relative pronoun.

3.1.4 Formation rules for Intermediate Structures

1. An Intermediate Structure representing a sentence is called a sentential Intermediate Structure (SIS).
Any well-formed Intermediate Structure representing a sentence has a verb node as its root.

2. An Intermediate Structure with an Argument node as root is called an Argument Intermediate Structure (AIS).
An Intermediate Structure representing a nominal is an AIS.
3. If s is a SIS and a is an AIS, then s' is a well-formed SIS, if s' is constructed from s and a by attaching a as last element to the list of ARGUMENTs of the RELATION in the root of s and defining the role name of the ARGUMENT forming the root of a.
4. If n and m are AIS, then n' is a well-formed AIS, if the root node of n contains a RELATION and m is attached to its list of ARGUMENTs and a role name is defined for the ARGUMENT forming the root of m.
5. If s is a SIS and a is an Argument node, then a' is an AIS, if s is attached to a and the argument type is set to VERBSTR.
6. If a and b are AIS and c is an Argument node of type COORD, then c' is an AIS if the contents of a is attached as left part of COORD, the contents of b is attached as right part of COORD, and the conjunction operator is defined.

3.2 The construction of Intermediate Structures from parse trees

To cover the Kamp fragment the following interpretation routines are needed:
PRNAME and ***NOMEN*** which map strings of characters to elements of AIS;
NPDEF, NPINDEF and ***NPQUAN*** which map pairs consisting of strings of characters and elements of AIS to elements of AIS;
VERB which maps strings of characters to elements of SIS;
NOM and ***ACC*** which operate according to Intermediate Structure formation rule 3;
RELCL which applies Intermediate Structure formation rule 5 and then 4;
COND which combines a pair of elements of SIS by applying Intermediate Structure formation rule 5 and then rule 3;
STMT which maps elements of SIS to DRSs.

These routines are applied as indicated in the parse tree and give the desired Intermediate Structure as a result.

4 Discourse Representation Structures

In this section we give a brief description of Kamp's Discourse Representation Theory (DRT). For a more detailed discussion of this theory and its general ramifications for natural language processing, cf. the papers by Kamp (1981) and Guenthner (1984a, 1984b).

According to DRT, each natural language sentence (or discourse) is associated with a so-called Discourse Representation Structure (DRS) on the basis of a set of DRS formatior rules. These rules are sensitive to both the syntactic structure of the sentences in question as well as to the DRS context in which in the sentence occurs.

4.1 Definition of Discourse Representation Structures

A DRS K for a discourse has the general form K = <U, Con> where U is a set of "discourse referents" for K and Con a set of "conditions" on these individuals. Conditions can be either ***atomic*** or ***complex***. An atomic condition has the form P(t1,...,tn) or t1=c, where ti is a discourse referent, c a proper name and P an n-place predicate.

Of the complex conditions we will only mention "implicational" conditions, written as K1 IMP K2, where K1 and K2 are also DRSs. With a discourse D is thus associated a Discourse Representation Structure which represents D in a quantifier-free "clausal" form, and which captures the propositional import of the discourse.

Among other things, DRT has important consequences for the treatment of anaphora which are due to the condition that only those discourse referents are admissible for a pronoun that are accessible from the DRS in which the pronoun occurs (A precise definition of accessibility is given in Kamp (1981)).

Discourse Representation Structures have been implemented by means of the three relations ASSERTION, ACCESSIBLE, and DR shown in the appendix. These three relations are written out to the relational database system (Astrahan &al (1976)) after the current text has been processed.

4.2 From Intermediate Structures to DRSs

The Intermediate Structures are processed starting at the top. The transformation of all the items in the Intermediate Structure are relatively straightforward, except for the proper semantic representation of pronouns. According to the spirit of DRT, pronouns are assigned discourse referents accessible from the DRS in which the pronoun occurs. In the example given in the appendix, as we can see from the ACCESSIBLE table there are only two discourse referents available, namely u1 and u2. Given the morphological information about these individuals the pronoun "it" can only be assigned the discourse referent u2 and this is as it should be. For further problems arising in anaphora resolution in general cf. Kamp (1981) and Guenthner and Lehmann (1983).

5 Remarks on work in progress

We are at present engaged in extending the above construction algorithm to a much wider variety of linguistic structures, in particular to the entire fragment of English covered by the USL grammar. Besides incorporating quite a few more aspects of discourse structure (presupposition, ambiguitity, cohesion) we are particularly interested in formulating a deductive account for the retrieval of information from DRSs. This account will mainly consist in combining techniques from the theory of relational database query as well as from present techniques in theorem proving.

In our opinion Kamp's theory of Discourse Representation Structures is at the moment the most promising vehicle for an adequate and efficient implementation of a natural language processing system. It incorporates an extremely versatile discourse-oriented representation language and it allows the precise specification of a number of up to now intractable discourse phenomena.

References

Astrahan, M. M., M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P. P. Griffiths, W. F. King, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, V. Watson (1976): "System R: Relational Approach to Database Management", *ACM Transactions on Database Systems*, vol. 1, no. 2, June 1976, p. 97.

Bertrand, O., J. J. Daudenarde, D. Starynkevich, A. Stenbock-Fermor (1976): "User Application Generator", *Proceedings of the IBM Technical Conference on Relational Data Base Systems*, Bari, Italy, p. 83.

Guenthner, F. (1984a) "Discourse Representation Theory and Databases", forthcoming.

Guenthner, F. (1984b) "Representing Discourse Representation Theory in PROLOG", forthcoming.

Guenthner, F., H. Lehmann (1983) "Rules for Pronominalization", *Proc. 1st Conference and Inaugural Meeting of the European Chapter of the ACL*, Pisa, 1983.

IBM (1981): *User Language Generator: Program Description/Operation Manual*, SB10-7352, IBM France, Paris.

Kamp, H. (1981) "A Theory of Truth and Semantic Representation", in Groenendijk, J. et al. *Formal Methods in the Study of Language*. Amsterdam.

Lehmann, H. (1978): "Interpretation of Natural Language in an Information System", *IBM J. Res. Develop.* vol. 22, p. 533.

Lehmann, H. (1980): "A System for Answering Questions in German", paper presented at the 6th International Symposium of the ALLC, Cambridge, England.

Ott, N. and M. Zoeppritz (1979): "USL - an Experimental Information System based on Natural Language", in L. Bolc (ed): *Natural Language Based Computer Systems*, Hanser, Munich.

de Sopeña Pastor, L. (1982): "Grammar of Spanish for User Specialty Languages", *TR 82.05.004*, IBM Heidelberg Scientific Cente-.

Zoeppritz, M. (1984): *Syntax for German in the User Specialty Languages System*, Niemeyer, Tübingen.

Appendix: Example

```
SENT
|
SC
|
+-------------------------------+-----+
|                               |     |
NP                              |     |
|                               |     |
+-----+                         |     |
|     |                         |     |
|     NOMEN                     |     |
|     |                         |     |
|     +------+                  |     |
|     |      |                  |     |
|     |      SC                 |     |
|     |      |                  |     |
|     |      +---+----+         |     |
|     |      |   |    |         |     |
|     |      |   |    NP        |     |
|     |      |   |    |         |     |
|     |      |   |    +------+  |     |
|     |      |   |    |      |  |     |
QU    NOMEN  NP  VERB QU NOMEN  VERB  NP
|     |      |   |    |  |      |     |
every farmer who owns a  donkey beats it
```

Parse tree

```
R: BEAT
   A(NOM): R: FARMER (EVERY)
              A(NOM): R: OWN
                         A(NOM): RELPRO
                         A(ACC): R: DONKEY (A)
   A(ACC): PERSPRO
```

Intermediate Structure

ASSERTION table

DRS#	ASSERTION
1	FARMER(u1)
1	OWN(u1,u2)
1	DONKEY(u2)
2	BEAT(u1,u2)

DR relation

DR	DRS#	Congr	S#	Level
u1	1	he	1	1
u2	1	it	1	2

ACCESSIBLE relation

upper DRS	lower DRS
1	2