Dealing with Cross-Sentential Anaphora Resolution in ALEP

Thierry Declerck
IMS, University of Stuttgart
Azenbergstr. 12
D-70174 Stuttgart
thierry@ims.uni-stuttgart.de

Abstract

The experiments described here have been done in connection with the LS-GRAM project, which is concerned with the development of large scale grammars and thus foreseen the coverage of "real life texts". But in order to deal with such texts, it is also necessary to process linguistic units which are larger than sentences. The resolution of cross-sentential anaphora is one of the problems we have to deal with, when we switch towards the analysis of such larger linguistic units. In order to propose an analysis of the cross-sentential anaphora, one has to be able to refer back to an antecedent, which is to be found in a preceding sentence. This will be done on the basis of an *information-passing* framework. Using also the simple unification technique a resolution of the pronoun can then be tried out: parts of the content information of the pronoun are going to be compared (unified) with specific parts of the content information of the (possible) antecedent.

1 Introduction

The experiments described below have been done in connection with the LS-GRAM project[1], which is concerned with the development of large scale grammars. The specifications of the project foreseen the coverage of "real life texts", which have also been processed by a corpus analysis. The results of the corpus analysis allowed us to determine a priority list of the linguistic phenonema to be described. And in order to deal with "real life texts", it is also necessary to consider the processing of linguistic units, which are larger than sentences. And as it is well known, the interpretation of sentences embedded in larger units is often distinct from the one of sentences, which are standing on their own.

[1]The LS-GRAM (Large-Scale GRAMmars for EC languages) project is funded by the CEC under the number LRE 61029. The examples and the grammar descriptions I am using are taken from the German grammar, see (Rieder & al. IAI).

The resolution of cross-sentential anaphora is one of the problems we have to deal with, when we switch towards the analysis (or synthesis) of such larger linguistic units. In order to give a correct interpretation of the cross-sentential anaphora, one has to be able to refer back to an antecedent, which is to be found in a preceding sentence, and I am using the term *information-passing* exactly in this sense: some information about a possible antecedent must be stored in order to be *passed on* to following sentences and to allow the anaphoric link, if some of the subsequent sentences are containing an anaphoric pronoun. Using the simple unification technique, as for the processing of other linguistic phenomena within ALEP, a resolution of the pronoun can then be tried out: parts of the content information of the pronoun are going to be compared (unified) with specific parts of the content information of the (possible) antecedent.

In the next section I will first show how larger linguistic units can be processed within the ALEP system. In a second section I will very briefly present a semantic framework, which introduces the idea of "information-passing" in order to cope with cross-sentential anaphora: the Dynamic Predicate Logic (DPL). In the last section I will show how a very preliminary and tentative implementation of this framework can be modelled within the ALEP formalism. Even if this first implementation is somehow primitive, this will permit us to formulate some remarks about the allowed degree of modularity of grammar descriptions within ALEP and also about the way in which such descriptions can be extended. These are two important aspects if one considers the task of developing large scale grammars.

2 The Text Handling System and the 'Paragraph unit'

Everyone who writes a grammar within the ALEP platform has some 'contact' with its Text Handling (TH) system, which converts each input into a SGML tagged expression. The TH component is the first processing step provided for by the ALEP system. In this tool, the *sentence* is defined as

the default linguistic unit. If larger units are to be processed, this has to be explicitely defined by the user. In our case, the linguistic unit is defined to be 'P' (for 'Paragraph')[2]. The output of the TH being for example:

```
<P>
 <S>
  <W>John</W>
  <W>sleeps</W>
   <PT>.</PT>
 </S>
</P>
```

one can refer to the tag 'P' in order to define this structure being the linguistic unit to be processed.

As usual (and also obligatory) for the development of grammars within ALEP, a so-called ts_ls_rule (a mapping between text structures and linguistic structures) has to be defined:

```
ts_ls_rule(
  ld:{
    ...
       syn => syn:{
         constype => phrasal:{
                       max => yes,
         constr => paragraph} } },
    'P', [] ).
```

where a linguistic description ('ld') defining the 'constr(uction)' type of a 'paragraph' is associated with the tag 'P', symbolizing the text type 'paragraph'. The distinguishing value here is 'paragraph', which has been added to the *type system* as a possible value for the feature 'constr': the ALEP formalism being type based, every feature, with its range of possible values, has to be declared in the declaration component.

The next step involves in the description of grammar rules which parse the structure of a paragraph. The phrase-structure rule responsible for the building of the paragraph-structure is simple. The mother node simply allows a binary branching of two sentential daughters. A recursion is defined on the right daughter, the value of 'constr' being a disjunction of 'punct_att' (describing a sentence terminated by a full stop) and 'paragraph' (describing thus the recursion). The left daughter is considered to be the head (structure-sharing of 'head' features), as one can see in the following (simplified) presentation of the rule:

```
ld:{
    ...
    syn => syn:{
       constype => phrasal:{
          constr => paragraph},
       category => head_cat:{
          head => HEAD => v_head:{} } } } }
< [
ld:{
    ...
    syn => syn:{
       constype => phrasal:{
           constr => punct_att},
       category => head_cat:{
         head => HEAD => v_head:{} } } } } ,
ld:{
    ...
    syn => syn:{
       constype => phrasal:{
         constr =>  (punct_att ; paragraph) },
       category => head_cat:{
          head => v_head} } } } ].
```

where '<' symbolizes the immediate dominance relation between the mother and the list of daughters.

In principle, these are the steps which are necessary in order to extend the coverage of the grammar to larger linguistic units. There is naturally some more technical work to be done, but this will be described in the third chapter, where I will go into more details of the architecture of the grammar development. At this stage, we are able to parse a paragraph and to get a syntactical analyse of this structure. Some aspects which are specific to text linguistic should be considered. The one I am concentrating on is the cross-sentential anaphoric relation. This has been postponed to the semantic which is treated within the *refinement* component of the grammar. But before explaining the motivation of the grammar design on this point and the reasons for postponing the semantic untill the process of refinement, the semantic framework which has been choosen for the modelling of the cross-sentential anaphora should be presented.

3 DPL as Representation Language for Information-Passing

The Dynamic Predicate Logic (DPL) results from an investigation of a dynamic semantic interpretation of the language of first order predicate logic and is "intended as a first step toward a compositional, non-representational theory of discourse semantics"[3]. This approach is concerned among other things with the cross-sentential anaphora. The dynamic aspect resides in the fact that, for this approach, the meaning of a sentence doesn't lie in its truth conditions, but "rather in the way it changes the ... information of the interpreter"[4]. DPL considers only the information change which concerns "their potential to 'pass-on' possible antecedents for subsequent anaphors"[5]. The Dynamic

[2] I would like to thank Gordon Cruickshank (Cray Systems, Luxembourg) who gave me the initial idea to use this strategy in order to describe the interdependency of information between sentences.

[3] I am refering here to (Groenendijk91).

[4] Ibid. p. 43

[5] Ibid. p. 44

Predicate Logic is based on the syntax of the standard predicate logic, but proposes a new (dynamic) interpretation of the quantifiers and connectives which allows the binding of variables within and outside their scope, depending on the interpretation of the corresponding expressions of the natural language.

Two (strong) assumptions, which are controversial in the discussion on this topic, are underlying the DPL approach: Indefinite NPs are considered to be *quantificational expressions* and pronouns to act like *variables*. Not everyone agrees on those assumptions, as this can be seen in the Discourse Representation Theory or in the work by Irene Heim[6]. But those assumptions are here important if one wants to provide an uniform translation of indefinite NPs into existential quantifier (see below). And the desired compositional treatments requires that the information concerning the pronouns is to be found in the sentences uttered so far, i.e. as included within the scope of a logical quantifier or connective.

The particular expressions of the natural language DPL is dealing with are the following:

(1) A man walks in the park. He whistles. – cross-sentential anaphora
(2) If a farmer owns a donkey, he beats it. – donkey sentence
(3) Every farmer who owns a donkey, beats it. – donkey sentence

And the problem consists in providing an adequate semantic representation of the anaphoric links. There are several ways of representing the semantic interpretation of each of the utterances and three of them (1 - 3) are discussed by Groenendijk & Stokhof:

(A) In classical predicate logic:
- $\exists x[man(x) \wedge walk\_in\_the\_park(x) \wedge whistle(x)]$ (1)
- $\forall x \forall y[[farmer(x) \wedge donkey(y) \wedge own(x,y)] \rightarrow beat(x,y)]$ (2) & (3)

(B) In a compositional way:
- $\exists x[man(x) \wedge walk\_in\_the\_park(x)] \wedge whistle(x)$ (1)
- $\exists x[farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]] \rightarrow beat(x,y)$ (2)
- $\forall x[[farmer(x) \wedge \exists y[donkey(y) \wedge own(x,y)]] \rightarrow beat(x,y)]$ (3)

(C) In the Discourse Representation Theory:
- $[x][man(x), walk\_in\_the\_park(x), whistle(x)]$ (1)
- $[\_][[x,y][farmer(x), donkey(y), own(x,y)] \rightarrow [\_][beat(x,y)]]$ (2) & (3)

What is missing in (A) is the compositional representation of the subparts of the utterances. Another disturbing point is the distinct translation of the indefinite NPs into the representational language, once as an existential quantifier (A,1) and once as an universal quantifier (A,2 & A,3). The fact that (2) and (3) translate into the same semantic representation is also reflecting the non-compositionality of the classical predicate logic in this case.

The problems with the compositional representation (B) are concerning the binding of the variables (the pronouns in the natural language). In (B,1) the third occurence of the variable x is free and thus doesn't allow the anaphoric reading. The same remarks are valid for x and y in (B,2) and for y in (B,3). But the way (B) is representating the utterances allows the uniform translation of indefinite NPs into an *existential quantifier*.

The problems with the DRT representation are more of methodological nature, since on the treatment of those cases, DPL and DRT are empirically equivalent. In short: Groenendijk and Stokhof are missing the compositional building of the semantic representation and also would prefer to use a more classical representational language, like the one of first order logic. For this, they are 'merging' together the representation (A) and (B), and considering now only the first case (1), the dynamic semantic interpretation is going to be like (B,1):

$$\exists x[man(x) \wedge walk\_in\_the\_park(x)] \wedge whistle(x),$$

but with the existential quantifier having scope over the conjunction of the two sentences, this representation is going to be equivalent to:

$$\exists x[man(x) \wedge walk\_in\_the\_park(x) \wedge whistle(x)].$$

This is possible because the interpretation of a sentence doesn't lie in a set of assignments, but rather in a set of ordered pairs of assignments, where those pairs represent the *input-output* states of a sentence. In our example, the first sentence has an output which is as the same time the input of the second one. Since the existential quantifier is interpreted as being able to quantify outside its scope (also in combination with the *conjunction* and the sequencing of sentences), the information concerning the (possible) antecedent is going to be passed-on to following sentences, which could be subsequently uttered. The fact that the existential quantifier in DPL is interpreted as a quantifier which can bind outside of its syntactic scope allows to say that we provide a compositional treatment of the utterance, the second sentence being interpreted as it comes, without referring to some metalinguistical representation or process. The existential quantifier is qualified as an *externally dynamic quantifier*.

Not every quantifier (or connective) has the dynamic property of binding outside of its scope; the universal quantifier, for example, can bind within its scope, but not outside of it:

(4) *Every man walks in the park. He whistles

[6] See (Kamp 1981) and (Heim 1982, p. 122)

is ruled out. The dynamic semantic interpretation of this quantifier blocks the passing of the information: the ouptut of the first sentence is empty (with respect to the information concerning anaphoric binding). The input of the following sentence will therefore contain no information allowing a resolution of the pronoun.

The way DPL is interpretating the distinct quantifiers and connectives is the following one:

- Existential quantification and conjunction are *externally* dynamic.
  They can bind variables within and outside their scope:
  [A man]$_i$ walks in the park and he$_i$ whistles.
  He$_i$ is happy
- Universal quantification and implication are *internally* dynamic.
  They can bind variables only inside their scope:
  Every farmer who owns [a donkey]$_i$, beats it$_i$
  *[Every man]$_i$ walks in the park. He$_i$ whistles
  If [a farmer]$_i$ owns [a donkey]$_j$, he$_i$ hates it$_j$
  *If [a farmer]$_i$ owns [a donkey]$_j$, he beats it.
  He$_i$ hates it$_j$
- Negation and disjunction are *static*.
  They cannot bind variables (at least, they don't allow a anaphoric reading):
  *[No man]$_i$ walks in the park. He$_i$ whistles
  *[A man]$_i$ walks in the park or he$_i$ whistles

This is too simple and for some English examples it seems to be wrong. The authors are considering and discussing the cases which contradict the assumptions and give some hints in order to integrate those cases. I will not discuss this point here, but just mention, that for the German grammar we should have a look at a detailled analysis of the meaning of such expressions[7]. Once this has been done, we can encode this information in the lexicon (as will be seen in the next section). But here we can say that the DPL approach allows us, to a certain degree, to account for the resolution of anaphora without having to leave the field of linguistic descriptions. With the only means of the grammar and the formalism we have, we are able to provide a first and simple description of those phenomena. It is still to be investigated how sophisticated such a treatment can be.

4 A first Implementation of the Dynamic Interpretation

As we have seen, the 'paragraph' has been defined as the linguistic unit to be processed by the system. To provide a (simple) syntax was so far not a problem. But, as stated in the second section, if a free pronoun occurs in a sentence, it is possible that this pronoun requires an anaphoric interpretation. To achieve this interpretation, some information about the antecedent is necessary and this information is to be found in a precedent sentence. DPL theory provides us with an elegant framework, describing the semantic of utterances as the way in which information is passed-on between sentences and so controlling the possible binding of pronouns.

4.1 The Organisation of the Lingware

I have tried to model the DPL framework within the ALEP platform. This experiment is documented in the following section. In doing this, I followed the overall strategy of the grammar development within ALEP. The syntactic 'paragraph' rule has been described within the *analysis* component of the grammar. The process of *analysis* is a process concerned with the building of structure trees induced by the ps-rules. In our grammars, this process is associated with a subpart of the lexicon, which is containing only the information relevant for that kind of process, i.e. the building of a parse tree, which is only dealing with morpho-syntactical information. One of the motivation of this organisation of the lexicon (and also of the rules associated with it) lies in the consideration of efficiency. At least for one of the parser of the system (the bottom-up head-out parser), the presence of multiple entries for one item and the description of more than one rule for a phenomenon has very negative consequences on the run-time behaviour of the system, backtracking being very expensive. But even if another parser (the record parser, which is not so sensitive to this kind of problems) is used, the use of shallow linguistic descriptions in the analysis component allows one to formulate some generalizations. Lexical and semantic ambiguities are than resolved or introduced at the following level, the *refinement* component, which corresponds to a process of decorating already existing trees: feature structures are just added to the trees. The refinement process is a special feature of the ALEP formalism. This process consumes very little cpu-time. For this process, specific linguistic subcomponents are described.

There is a relation between the sublexicons, this relation being one of subsumption. Linguisitic descriptions contained in the distinct sublexicons have all the same structure but the degree of specification is different from one lexicon to another. Thus we don't have a stratificational model and the descriptions stay declarative and monotonic. We are just describing what subparts of information of an entry is going to be accessed by a process. Since this organisation of the lexicon (and the associated rules) is done along the line of the processes provided by ALEP, we call it the *vertical* organisation.

The description of this vertical organisation is done with the help of the 'specifier features' provided

[7] As for example in (Bethke 1990) or in (Vater 1979).

by the ALEP system, wich are configurable by the user. It is in such a way possible to determine what subparts of the grammar are going to be accessed by a particular process.

4.2 The Lexicon Entries and the Rules for the Resolution of Anaphora

The rule building the paragraph structure introduced before produces just one tree. The possible distinct readings of it are described within the refinement component. Thus the cross-sentential anaphora relation (being essentially a semantic process) is fully described on this level (the morpho-syntactical aspects being described in the analysis component of the grammar).

We will no see how the relevant items are described in the *refine* lexicon. I just consider here the entries of substantives, pronouns and quantificational expressions (determiners). My goal is to provide the informations that are necessary for the modelling of the dynamic treatment of the cross-sentential anaphora. Here the way they are coded in the German grammar:

- Referential expressions

```
ld:{
....
content  => ...
   restr  =>
     [inst_zero_psoa:{
          rel  => rel:{
               rel_name  => flugzeug },
          inst     => A48 } ],
      indx  => A48 => ind_indx:{
                pers          => p3,
                numb          => sing,
                gend          => neut }
 } } }.
```

- Pronominal expressions

```
ld:{
...
content  => ...
    restr  =>
       [inst_zero_psoa:{
           rel  =>  rel:{
             rel_name  => PRO },
           inst      => A48 } ],
     indx  => A48 => ind_indx:{
               pers          => p3,
               numb          => sing,
               gend          => neut }
 } } } }.
```

- Quantificational expressions

```
det_d_das ~
mLDref_core[
  det_d_das,
  das,
  mLA_sem_funct_sem_det[_] ].

det_ein~
mLDref_core[
  det_ein,
  ein,
  mLA_sem_funct_sem_det[extern_dynamic] ].

det_kein~
mLDref_core[
  det_kein,
  kein,
  mLA_sem_funct_sem_det[static] ].

det_d_jedes ~
mLDref_core[
  det_jedes,
  jedes,
  mLA_sem_funct_sem_det[intern_dynamic] ].
```

I encoded the information about semantic gender and number of the referential expressions and the pronouns (contained in the 'restr(iction)' attribute. The 'rel_name' attribute represents the referential property of the item: this a variable ('PRO') in the case of the pronouns, modelling in this way the DPL assumption that pronouns are acting as variables. The entries of quantifiers and determiners are presented here as macros. The relevant information for us is the one concerning the quantificational force of this entries (I don't consider here binary conjunctives) which is lexically determined. In case of 'das' (the), the relevant information has been left unspecified, since where are not considering definite descriptions for the time being. The quantificational force of 'ein' (a, an) has been specified as 'extern_dynamic', the one of 'kein' (no, no one) as 'static' and the one of 'jedes' (every) as 'intern_dynamic', modelling the classificiation proposed in the DPL framework.

During the processing of the paragraph, if a free pronoun occurs in a sentence and if it can refer to an antecedent, the value of the 'quantificational_force' of the antecedent should be 'extern_dynamic' and the values of the attribute 'restr' of the antecedent and the pronoun must be unifiable. In this case, the value of the 'rel_name' of the pronoun is unified with the value of the one of the antecedent, as one can see below, where in this case the values of the 'restr' features are variable-shared. Here a (simplified) rule accounting for resolution in the context of one-argument predicates (the other cases are described by rules disjunction):

```
ld:{
....
sem => sem:{
  content => lq_cont:{
  ....
     arg1 =>  lq_cont:{
       quants => [quantifier:{
          q_force => Q1 } ],
             rd_cont => r_indx:{
                restr => RESTR1 } } },
    ....
```

```
      arg1 =>  lq_cont:{
        quants => [quantifier:{
           q_force => Q2 }],
               rd_cont => r_npro:{
                    restr => RESTR2 } } }  ]
  } } } }
< [
ld:{ ....
sem => sem:{
  content => lq_cont:{
   ....
    arg1 =>  lq_cont:{
      quants => [quantifier:{
        q_force => Q1 } ],
          rd_cont => r_npro:{
               restr => RESTR1 } } } } } } },
ld:{ ....
sem => sem:{
  content => lq_cont:{
   ....
    arg1 => ( lq_cont:{
      quants => [quantifier:{
        q_force => Q1 => extern_dynamic} ],
          rd_cont => r_ppro:{
               restr => RESTR1 } }
    /  quants => [quantifier:{
        q_force => Q2} ],
          rd_cont => r_npro:{
             restr => RESTR2 } } } } } } ].
```

With this simple technique, we are able to accept "Ein Mann kommt. Er singt" and to reject "Jeder Mann kommt. Er singt" – the value of 'quantificational_force' of the entry 'jeder' (every) is 'intern_dynamic' and so disallows the unification of the 'restr' values. The same with "Kein man kommt. Er singt". The negation is a static semantic phenomenon and the value of its attribute 'quantificational_force' is 'static'.[8] If there is no pronoun in the second sentence, no unification is tried out: the values of the corresponding 'restr' attributes are not structure-shared. The building of the paragraph structure just goes on. The values of the attributes 'quantificational_force' and 'restr' are then put together in a list.

4.3 Modularity and Extendability

The short experiment described in this paper is really too primitive and doesn't allow any statement about the possibility of providing a complete treatement of the cross-sentential anaphora on the basis of the DPL framework. But one goal of the experiment was also to gain some knowledge about the possible extension of the coverage of the grammar. And heresome conclusions can be drawn.

[8] Linguisitically speaking, this is too simple. In German at least, negated NPs can often bind pronouns. And we should also allow *generic* readings. Some experiments have been done with respect to this. But actually, I just would like to show how information-passing can be modelled in ALEP.

First of all, the modular organisation of the grammar development within ALEP proved itself to be very practical. The 'tools' provided by ALEP (the 'specifier features'), if they are reasonably configurated during grammar development, allow a high degree of modularity and permit, without difficulties, to define new grammar components.

With the help of the text handling component, it was also no problem to extend the coverage of the grammar to larger linguisitc units. The description of grammar components for such units is well supported.

It still remains the task of providing some preference descriptions for anaphora resolution. This will be done along a more detailled linguistic analysis and considering corpus analysis. ALEP provides also for constraint solvers which allow to define such preferences in an elegant (propositional) way.

References

BIM/SEMA, *ALEP System Documentation*, CEC 1993, Luxembourg.

Simpkins, N.K.; Groenendijk, M. and Cruickshank G. (P-E International), *ALEP 1 User Guide*, CEC 1993, Luxembourg.

Badia, T.; Bredenkamp, A.; Declerck, T.; Hentze, R.; Marimon, M.; Schmidt, P.; Theofilidis, A. (1995), LS-GRAM Rule Coding Manual, Deliverable D - WP7 Version 1, CEC, 1995, Luxembourg.

Bethke, I. *der die das* als Pronomen, 1990.

J. Groenendijk & M. Stokhof, Dynamic Predicate Logic, in *Linguistics and Philosophy* 14, p. 39-100, 1991

Heim, Irene, *The Semantics of Definite and Indefinite Noun Phrases*, dissertation, University of Massachussetts, Amherst, 1982.

Kamp, Hans. A Theory of Truth and Semantic Representation. In J. Groenendijk, T. Janssen, and M. Stokhof (eds.), *Formal Methods in the Study of Language*,, Mathematical Center Amsterdam: 277-322, 1981 [reprinted in J. Groenendijk, T. Janssen, and M. Stokhof (eds.), *Truth, Interpretation and Information*, Foris, Dordrecht, 1-41, 1984.

Sibylle Rieder, Paul Schmidt & Axel Theofilidis - IAI, & Thierry Declerck - IMS. *LS-GRAM Lingware*, Documentation, Deliverable C-D-WP6e, 1996, Luxembourg.

Vater, Heinz. *Das System der Artikelformen im gegenwaertigen Deutsch.* 2. edition, Tuebingen, 1979.