CROSSING COREFERENCE IN DISCOURSE REPRESENTATION THEORY

Michael HESS
University of Zurich
Dept. of Computer Science, Winterthurerstr. 190
CH-8057 Zurich, Switzerland

Abstract

Sentences with crossing coreference (Bach-Peters-sentences) are notoriously difficult to explain in a natural manner. An intriguing parallel with certain properties of Prolog suggests a modification to Discourse Representation Theory which allows a simple and coherent explanation of these, and related, sentences.

The Problem

In English there is one type of sentence that has caused major problems for practically all linguistic theories that have tried to explain it, and none of the explanations put forward is very convincing. The sentences in question are those with *crossing coreference*, the so-called *Bach-Peters-sentences*. The standard examples are

1) `The hunter who shot at it hit the lion that chased him`

and, with explicit quantifier expressions:

2) `Every man who wants it will get the prize he deserves`

What is the difficulty with this type of sentence? They contain two noun phrases each of which contains a pronoun that refers to the other noun phrase, and the first pronoun is furthermore a case of "backwards anaphora", or "cataphora". These sentences are admittedly rare, but sentences with simple (non-crossing) cataphora are quite frequent in real world English (Carden 1982). And Bach-Peters-sentences are nevertheless perfectly regular, and so they should find a natural explanation. Moreover, they are key examples of sentences where cataphora cannot, in principle, be replaced by anaphora (cf. also Mittwoch 1983). This is important because one of the standard approaches to cataphora has been to define it away as stylistic variant of anaphora, which can be "rectified" by a simple transposition. In other words: Since we have to find a way to explain cataphora for Bach-Peters-sentences anyway, we can save us the trouble to devise such tricks for the simpler cases.

Are Bach-Peters-Pronouns Descriptional Pronouns?

The non-reducibility of cataphoric to anaphoric pronouns in Bach-Peters-sentences becomes clear if we try to explain them in the traditional manner. It seems that both of the two traditional interpretations of pronouns, the "descriptional" as well as the "denotational" one, fail to explain the intuitive truth conditions of Bach-Peters-sentences. In Transformational Grammar the descriptional approach is taken, and pronouns are always expanded to the *surface syntax form* of the noun phrase they anaphorically refer to (in other words, pronominalization is an obligatory cyclic rule). But then we get, for the example above, a double *infinite embedding* of relative clauses:

```
The hunter who shot at
  (the lion that chased
     (the hunter who shot at
        (the lion that chased ...)))
hit the lion that chased
   (the hunter who shot at
     (the lion that chased
        (the hunter who shot at ...)))
```

This analysis is patently useless. Karttunen shows (Karttunen 1971) that dropping the requirement that pronominalization is a cyclic rule alleviates the problem somewhat, but at a cost: It would make sentence 1 derivable from (at least) two different deep structures, viz. from the deep structures corresponding to the sentences

3) `The hunter who shot at the lion that chased him hit it`

4) `The lion that chased the hunter who shot at it was hit by him`

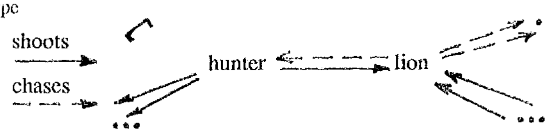
This would mean that 1 has to be ambiguous between the meanings of 3 and 4. This is what Karttunen assumes, but it is a highly dubious claim as Karttunen himself seems to feel (Karttunen 1971:167 f.). Moreover, the structure of 1 could also be derived from a deep structure corresponding to

5) `The hunter who shot at the lion hit the lion that chased the hunter`

But this sentence is considered by many informants to be simply ungrammatical (Karttunen 1971:178), and it is not acceptable at all under the coreference relations that should obtain between the noun phrases. Finally, the assumption that 1 is three ways ambiguous between 3, 4 and 5 is unacceptable, too. In order to show this, and in order to understand better what these three sentences really mean, we can use a set of data bases (after Karttunen 1971) which either contain, or do not contain, well-defined referents for the various definite noun phrases occurring in the example sentences. We will see that 1 is *not* ambiguous between 3 and 4, but that the three sentences have three distinct meanings which can be derived directly from their syntactic structure. This proves, at the same time, that cataphoric pronouns are irreducible in Bach-Peters-sentences. Let us first consider the definite noun phrase "the hunter who shot at the lion that chased him" (from 3) consisting of an embedding of two definite noun phrases. Since each singular definite noun phrase presupposes that there is a unique referent for it, this phrase can refer to a pair "hunter H - lion L" in the case where lion L is the only lion chasing hunter H, and this hunter H shot at this very lion L. In the following data base (Karttunen 1971:166) there is one such pair.

```
hunter(h1).          chased(l1,h3).       lion(l1).
hunter(h2).          chased(l2,h1).       lion(l2).
hunter(h3).          chased(l3,h2).       lion(l3).
shot_at(h1,l1).      shot_at(h2,l1).      shot_at(h2,l3).
shot_at(h1,l3).      shot_at(h3,l3).      shot_at(h3,l2).
```

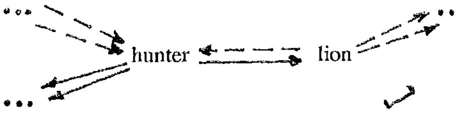
For each hunter, there is a single lion chasing him but only one hunter also shoots at this lion, viz. hunter 2 who shoots at lion 3. Hunter 2 shoots at other lions, as well (e.g. at lion 1) but lion 1 doesn't chase hunter 2 (although it does chase other hunters, e.g. hunter 3). Hence it can be said (Dik 1973:320) that the definite noun phrase "the hunter who shot at the lion that chased him" has a well-defined referent in any data base which contains *just one* configuration of the type

6) 

This rules out that other lions chase the hunter, but it leaves open the possibility that the hunter shoots at other lions, that the lion chases other hunters and, in particular, that other hunters shoot at the lion. On the other hand, example 4, "The lion that chased the hunter who shot at it was hit by him", is not interpretable in the data base given above. Its subject, "the lion that chased the hunter who shot at it", fails to refer properly: There is only one lion for which there is a hunter we can call "*the* hunter who shot at it", viz. lion 2 (the other two lions are both being shot at by more than one hunter), but lion 2 does not chase "the hunter who shot at it" (viz. hunter 3), and so the entire noun phrase fails to refer. A data base where there *is* a referent for this noun phrase could look like that:

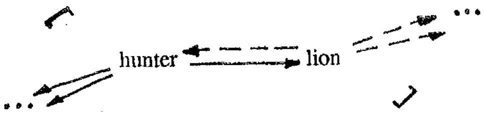
```
hunter(h1).          chased(l1,h1).       lion(l1).
hunter(h2).          chased(l1,h3).       lion(l2).
hunter(h3).          chased(l2,h1).       lion(l3).
shot_at(h1,l2).      chased(l2,h3).       chased(l3,h2).
shot_at(h2,l1).      shot_at(h3,l3).      chased(l3,h1).
```

The definite noun phrase "the lion that chased the hunter who shot at it" has a well-defined referent in any data base which contains *just one* configuration of the type 7:

7) 

Example 1 *cannot* be interpreted in *either* the first *or* the second data base. It is interpretable only in a data base which contains at most one configuration like 8 which *combines* restrictions 6 and 7:

8)



Are Bach-Peters-Pronouns "Bound Variable" Pronouns?

Since the unmodified as well as the modified descriptional interpretations of pronouns in Bach-Peters-sentences do not allow us to represent these distinctions they cannot be accepted. But how do the "deep" approaches to pronouns, the so-called "denotational" interpretations, fare?

The prototypical denotational interpretation of pronouns is the one suggested by First Order Logic, where bound variables are seen as the logical counterpart of pronouns in Natural Language. If we use iota-operators, we could try to translate 1 into

```
9)  hit(iota Z: [hunter(Z) ∧ shot_at(Z,W)],
        iota W: [lion(W) ∧ chased(W,Z)])
```

The trouble is that iota-operators bind their variables, making them inaccessible for reference from the outside. Hence we cannot refer forward from the term "shot_at(Z,*W*)" to the "W" in the second iota-expression, nor can we refer backward from "chased(W,*Z*)" to the "Z" in the first iota-expression. If we add equality to First Order Logic there is a way out. We could re-phrase 9 as

```
10) ∃ X: ∃ Y: (hit(X,Y) ∧
                X=iota Z: [hunter(Z) ∧ shot_at(Z,Y)] ∧
                Y=iota W: [lion(W) ∧ chased(W,X)])
```

The other examples, 3 and 4, would then become

```
11) ∃ X: ∃ Y: (hit(X,Y) ∧
                X= iota Z: [hunter(Z) ∧
                shot_at(Z,iota W: [lion(W) ∧ chased(W,Z)])] ∧
                Y= iota V: [lion(V) ∧ chased(V,X)])
12) ∃ X: ∃ Y: (hit(X,Y) ∧
                X= iota Z: [hunter(Z) ∧ shot_at(Z,Y)] ∧
                Y= iota W: [lion(W) ∧
                chased(W, iota V: [hunter(V) ∧ shot_at(V,W)])])
```

But in cases 11 and 12 part of the expression must be *repeated*, namely the one stating the uniqueness of the hunter-chasing lion, and of the lion-shooting hunter, respectively. This is a very unattractive way to express this sort of thing, and Karttunen agrees: "One is tempted to think that one of [these repeated definite descriptions] could be eliminated by a more clever use of variables, especially when variables in predicate calculus are generally used very much the same way as pronouns in natural language. [...] [But] the second appearance of the same description cannot be avoided, because, in predicate calculus, there is no way to refer back to the first." (Karttunen 1971:176).

McCawley's suggestion: Referential indices

But this is the point where other people disagree. McCawley (McCawley 1970), for instance, argues (for other reasons), that the semantic representation of sentences should not be cast in terms of First Order Logic, but rather in terms of the overall *proposition* of a sentence plus *referential indices*. The proposition would define the relationship that exists between the different objects talked about in the sentence, but these objects would be represented independently by referential indices that correspond to the "intended reference" of the noun phrase. These indices are "identified" by the noun phrases in the sentence, i.e. their values are determined by the noun phrases. A sentence such as "The man killed the woman", would be translated as

```
13) s(
        proposition(killed(X1,X2)),
        np(X1: the man)
        np(X2: the woman))
```

A surface sentence would be generated from this structure by *replacing* the referential indices in the proposition by the noun phrases identifying them. As for Bach-Peters-sentences, example 1 would be represented as

```
14) s(
        proposition(hit(X1,X2)),
        np(X1: the hunter who shot at X2)
        np(X2: the lion that chased X1))
```

Here the referential indices are identified by noun phrases which *themselves* contain referential indices. If we want to generate a surface sentence from this structure we will have to replace the referential indices by their identifying noun phrases, but we will not be able to replace systematically *all* referential indices this way, because this would lead to the same kind of infinite embedding that we encountered above. We will rather, at some point in the derivation, have to turn some of the referential indices into *pronouns* (taking into account case, gender and number). The point at which we stop replacing referential indices by full noun phrases and start turning them into pronouns will determine which of several possible paraphrases of a sentence we will obtain. The distinction between "proposition" on the one hand and "referential index" with the accompanying "identifying noun phrase" on the other hand allows McCawley to overcome the problem with repeated components: The "identifying noun phrase" of a referential index is mentioned only once, independently of how many times the referential index itself is used elsewhere in the representation.

Dik's Modification of McCawley's Theory

This approach has been criticized on different counts. *First*, it is not clear *at which point* referential indices may begin to be turned into pronouns rather than being replaced by identifying expressions. *Second*, and more importantly, McCawley's suggestion allows *all three* sentences (1, 3, and 4) to be derived from the same semantic representation, which would require that they are all synonymous. Hence all the problems we encountered with the descriptional interpretation of pronouns are back with a vengeance. Does that mean that we have to return to the standard First Order Logic representation with all its unattractive features (repetition of components)? Dik 1973 suggests a modification of McCawley's approach that takes care of the empirical fact that the three sentences mentioned above are not synonymous.

The main syntactic difference between the three example sentences is the way in which the different full noun phrases are embedded in each other. In particular, in the Bach-Peters-sentence there are *no* embedded *full* noun phrases; the only embedded noun phrases are pronouns ("who shot at *it*", "that chased *him*"). This kind of distinction is lost in McCawley's original notation, which is the reason why he has to claim that the examples sentences are all synonymous. Dik's first (and main) modification of McCawley's notation makes sure that this, crucial, distinction is not lost. The way to achieve this goal is by introducing what might be called (not Dik's expression) "annotated variables" to take the place of McCawley's referential indices, which are *constants*. The annotation of a variable indicates how the value of the variable is to be computed. Thus "X2(X2=iota Z: (lion(Z) ∧ chased(Z,Y)))" is an annotated variable. This would give us, for 3,

```
15) hit(X1,X2)
        X1=iota Y: (hunter(Y) ∧
        shot_at(Y,X2(X2=iota Z: (lion(Z) ∧ chased(Z,Y)))))
```

As in McCawley's system, referential indices must be replaced by the corresponding identifying expressions, and any variables remaining after this step are turned into pronouns. In addition, we have the usual convention that functional expressions are to be evaluated from the inside out. But what is gained by faithfully copying a certain syntactic structure (viz. embedding) into the deep representation? Are the interpretation rules given by Dik really sufficient to interpret the resulting expressions correctly? If we perform the replacement, for instance, in 15 we get

```
16) hit(iota Y: (hunter(Y) ∧
       shot_at(Y,X2(X2=iota Z: (lion(Z) ∧ chased(Z,Y))))),X2)
```

Now, if we wanted to generate a surface sentence from this representation we would turn the remaining unbound variable, i.e. "X2", into a pronoun, and everything would be in order. But what if we wanted to *evaluate* this statement over one of the data bases? The X2 on the level of the proposition, i.e. "hit(...,X2)" is clearly *outside* the scope of the "annotation" defining its value, and so we would expect it to remain unbound after the annotation itself has been evaluated. This is, after all, precisely the reason why Karttunen thought it necessary to re-introduce the duplication of logical expressions that McCawley had tried to get rid of: To provide the second, outer, occurrence of this variable with a value. We need an interpretation rule which specifies how variables of this kind *can* be bound, and this rule is not provided by Dik.

An Unexpectedly Simple Solution Suggested by Second-Order Prolog Constructs

As it turns out, the additional interpretation rule that makes the correct interpretation of Bach-Peters-sentences virtually "fall out" is the definition of second-order operators, and the general interpretation rules of Horn Clause Logic, as implemented in standard Prolog: Instead of "Z=iota X: (Y)" we use "setof(X,Y,[Z])", where the uniqueness requirement is built into the definition of "setof", and singularity is enforced by requiring the result list to consist of exactly one element. Annotated variables on the other hand are "multiplied out" in the relational spirit of Prolog, i.e. instead of "predicate(X,Y(Y=Z))" we write "(predicate(X,Y), Y=Z)". Combining these two steps we get, for the expression above,

```
17) hit(X1,X2),
        setof(Y, (hunter(Y),
                     setof(Z, (lion(Z), chased(Z,Y)), [X2]),
                             shot_at(Y,X2)), [X1])
```

or, with a more suggestive choice of variable names and a more efficient ordering of the goals

```
18) setof(H, (hunter(H),shot_at(H,TL),
                 setof(L, (lion(L),chased(L,TH)),[TL])),[TH]),
    hit(TH,TL).
```

Now the desired truth conditions come out correctly. We can see this if we treat 18 as a Prolog query: We find, first, a hunter ("H") who shoots at something ("TL"). Then we check whether this entity is identical with the set of exactly one lion ("[TL]") that chases someone ("TH") who must then turn out to be identical with the hunter who is the only such hunter ("[TH]"). Finally we check whether this hunter also hits this lion. The other sentences are represented the same way: 4 and 1 (the Bach-Peters-sentence) give

```
19) setof(L, (lion(L),chased(L,TH),
                 setof(H, (hunter(H),shot_at(H,TL)),[TH])),[TL]),
    hit(TH,TL).
20) setof(H, (hunter(H),shot_at(H,TL)),[TH]),
    setof(L, (lion(L),chased(L,TH)),[TL]),
    hit(TH,TL).
```

Now we get, without any additional stipulations, the three different interpretations for the three example sentences. The sentences are neither collapsed into one single meaning representation (with three synonymous surface sentences), as with McCawley's approach, nor into two different ones (with two distinct and unambiguous, and one ambiguous, surface sentence), as with Karttunen's approach. The simple fact that in Bach-Peters-sentences the full definite descriptions are not embedded, forces them to evaluate to two distinct, independently unique, values, and this ensures that these sentences are true *only* over data bases meeting condition 8.

But how can this unexpectedly straightforward solution be explained? The main problem that McCawley's representation, and Dik's modification of it, tried to overcome was: How can variable values be communicated into iota-expressions from the outside, despite the fact that, in First Order Logic, variables within the scope of an operator or quantifier are shielded from the outside? Now, in Horn Clause Logic all variables of a clause are, implicitly, universally quantified (which means that variables in a query, i.e. in a negated clause, are existentially quantified), and the *scope* of the implicit quantifiers is the *entire clause*. The bindings of variables "spread" throughout the clause, irrespective of how deep a variable may be embedded. This *also applies to the setof-operator*: All its variables are accessible from the outside, within the given clause. The variables can, of course, still be unbound when the evaluation of the setof-expression begins, and then it is the evaluation of the setof-expression that will establish bindings for these variables. But they can also get bound elsewhere, before the operator is used, and "spread forward". And then a proof of the setof-expression treats these pre-established bindings as *constraints* to be satisfied. This is the Prolog way to implement the cataphoric pronoun in Bach-Peters-sentences. In *this*, last, respect the setof-operator in Prolog is treated as just another *predicate*, and its being second order is irrelevant. The difference is that the *interpretation* of the setof-operator uses Prolog's *meta-call facility*. Or, to put it differently: A piece of code (the entries in the second argument of the setof-operator) is *first* treated as "*data*" (variable bindings are communicated with the outside world), and *then* it is treated as a piece of "*program*" that is executed (using the variable bindings that are established at this point in time). And in this the Prolog setof-operator differs fundamentally from the iota-operator as used in First Order Logic. The iota-operator has, as far as variable binding is concerned, the same force as a quantifier: A variable in its scope is immune from any outside interference.

Now it is clear why we need not repeat any expressions in the representation of the example sentences, and yet the variables all get properly bound: In the example above, the two terms of 18 (i.e. the setof-expression and the expression "hit(TH,TL)") are part of the same clause, and values for variables created in either of them will spread to the other. In particular, the value which the variable "TL" takes during the evaluation of "shot_at(H,TL)" is still available during the evaluation of the embedded "setof(L,(lion(L),chased(L,TH)),[TL])", and later during the evaluation of "hit(TH,TL)".

Mapping Variables onto Pronouns

If we want to generate surface sentences from these structures, we must distinguish between two uses of variables: First there are those uses which are merely an artefact of the relational way of representing functional application, and, second, there are those that correspond to true anaphoric relations in language. The first use is simple: If we want to represent functional applications such as

plus(times(3,2),4)

in a relational language, we must "multiply out" the embedded expression and create auxiliary variables for the intermediate results, e.g. "X" and "Y" in

times(3,2,X), plus(X,4,Y)

Thus we had to use 20, with auxiliary variables "TH" and "TL", instead of a functional representation such as, for instance,

```
21) hit(set(H, (hunter(H),shot_at(H,L))),
        set(L, (lion(L),chased(L,H))))
```

These "auxiliary" variables are situated on the same level of embedding (by definition: their purpose is to flatten embeddings). Co-occurrence of such variables on the same level of embedding maps, in simple cases, onto concatenation ("^") in surface structure: Thus the following occurrences of variables "TL" and "TH" in 20

```
20a) setof(L, ( ... ),[TL]),
     setof(H,  ( ... ),[TH]),
     hit(TH,TL).
```

become "((the hunter) ^ hit ^ (the lion))". Co-occurrence of variables *across different levels of embedding*, however, cannot be encoded as simple concatenation. These cross-references map onto pronouns. (The converse does not hold: There are pronouns that do not correspond to this kind of cross-reference; e.g. descriptional pronouns.) Thus the level-crossing co-occurrence of the variable "TL" in

```
20b) setof(H, ( ... shot_at(H,TL)), ...)
     setof(L, ( ... ),[TL]),
     hit(...)
```

must map onto a pronoun ("The hunter who shot at *it*"). A problem arises when we try to translate 19 back into English. If we begin the translation process with "hit(TH,TL)" and map the level-crossing variable "TH" onto a pronoun we get "He hit the lion that chased the hunter who shot at it", which is not acceptable under the intended interpretation (i.e. coreferentiality of "he" and "the hunter ..."). This corresponds to a well-known syntactic restriction on the use of cataphoric pronouns. Here we need a rule that works for syntax generation rather than for analysis. The following rule is a bit ad-hoc, but it is sufficient for the present purpose: We require that the translation of the entire set of expressions must begin with the expression defining the top relationship between the individual set expressions (i.e. with the expression corresponding, in most cases, to the main verb of the surface sentence), and that level-crossing occurrences of variables in this term are translated last. If this restriction makes it impossible to translate these variables from, say, left to right (as in the case of example 19, where the first variable "TH" is a level-crossing occurrence), it is done right to left. This requires that the surface verb form is passivized but it gives the grammatically correct ordering of full noun phrases and pronouns (i.e. we get the original passive sentence 4 for 19). In Bach-Peters-sentences such as 20 both the active and passive versions are admissible under this restriction, in keeping with the linguistic facts.

Second-Order Prolog Constructs and Discourse Representation Theory

The painless way in which the correct truth conditions of Bach-Peters-sentences and the related sentences virtually fall out of the standard Prolog interpretation rules and the definition of the second-order setof-operator is not just a lucky coincidence. It is rather another case of the intriguing parallel between Natural Language and Horn Clause Logic which has become particularly clear in Discourse Representation Theory (DRT). The main hypothesis of DRT is that *noun phrases* (and articles) have no quantificational force on their own but are *implicitly quantified* by the context. This allows DRT to explain, with remarkable ease, so-called donkey-sentences, a type of sentence that does not yield to the traditional interpretation of noun phrases as quantified statements. The correspondence between the logic underlying DRT, and Horn Clause Logic is, in this respect, almost one-to-one: In DRT, (indefinite) noun phrases introduce discourse referents which are quantified by the (discourse) context in the same way as variables in Horn Clause Logic are implicitly quantified by the (clause) context.

How do Bach-Peters-sentences fit into DRT? First, we notice the parallel between McCawley's ideas and DRT: His *"referential indices"* correspond, in their intended function, to the *discourse referents* in DRT, and *"propositions"* correspond to the DRT *"conditions"* on discourse referents. In the Prolog version of Dik's modification of McCawley's ideas, discourse referents correspond to the value of the third argument in a setof-operator, and the "conditions" of a Discourse Representation Structure (DRS) to the expression(s) in its second argument. All this applies, for the time being, only to *definite* noun phrases and their representation. If we want to incorporate this into DRT, we must first provide for the *possiblity to explicitly represent the embedding of noun phrases*. This kind of explicit embedding was the main reason why we got the right truth conditions in the Prolog representation of the example sentences. We must, in other words, be allowed to use embedded "conditions" in a DRS. Traditional DRT allows for the embedding of entire DRSs, but not of individual conditions. While a sentence like "If John owns a donkey *that dislikes him*, he beats it" is traditionally represented as a *flat* DRS like

```
22)  [u1:
      [u2: john(u1), donkey(u2), owns(u1,u2), dislikes(u2,u1)]
        →   [beats(u1,u2)]]
```

(cf. Kamp 1981, Kamp 1983, Frey 1983, Guenthner 1983, Guenthner 1985, Kolb 1985, Guenthner 1986, Pinkal 1986, Root 1986) this will not do for the sentences with embedded definite noun phrases considered above. We must somehow represent this embedding. And we must, obviously, provide for the interpretation rules to use them. These rules will crucially rely on Prolog's meta-call facility to implement the double use of embedded set-expressions, as data structures on the one hand and as "executable procedures" (i.e. as provable assertions) on the other.

In traditional DRT mostly indefinite and universal noun phrases (and proper names) are used while the Bach-Peters-sentences considered above all contained *definite* noun phrases. But for some of them there are versions with indefinite noun phrases, too, and all of them have corresponding plural versions. In order to cover *all* these cases we must introduce, instead of the "conditions" of DRT, *generalised set expressions* without the totality implication of Prolog's "setof" (cf. also Webber 1983). We use "set(Def,Card,Gdr,Var,Int,Ext)", where "Def" can take the values "def"(inite) or "indef"(inite), and "Card" either an explicit number, "plur", or a quantifying expression ("all", "some" etc.). "Gdr" gives the gender of the main noun. "Var"(iable),"Int"(ension) and "Ext"(ension) correspond to the three arguments of the setof-operator. The variable "Ext" can now stand for sets as well as for individuals.

Mapping Pronouns onto Variables

So far we have mentioned how pronouns correspond, statically, to certain semantic objects of our modified DRSs (i.e. to the level-crossing occurrences of variables). But it is one of the main goals of DRT to give a unified account of what the *procedures* that actually perform the resolution of pronouns should look like. This problem is much harder than the converse one, i.e. the mapping of level-crossing variables onto pronouns. The central idea used here by DRT is simple (it goes back to Karttunen, together with the term "discourse referent"): Indefinite noun phrases in "assertive" contexts create discourse referents which "live on", and which can be accessed by anaphoric expressions from points later in the sentence or discourse. Discourse referents, however, that are created by indefinites in universal, conditional, and negative contexts, "die off" when the sentence in which they occur is processed. This idea corresponds closely to Prolog's concept of variables and Skolem constants (the latter standing for existentially quantified variables): During the interpretation of a program variables remain accessible *by name* within the clause where they occur. This corresponds to the limited life-span of discourse referents created in universal, conditional, and negative contexts. For Skolem constants in Prolog, however, the scope is the entire program; they "live forever", in the same way as discourse referents created by indefinites in assertive contexts. And whenever a (definite) pronoun or definite noun phrase is encountered, a suitable antecedent must be located among the discourse referents still "alive". Its value is then replaced by the value of the discourse referent found. If several pronouns access the same discourse referent, they get, of course, the same value. This is the DRT counterpart of unification.

If we want to have, in our modified notation, discourse referents "float on the surface" of the corresponding DRSs, accessible for later anaphoric reference, we could write, for the *in*definite version of 3, viz. "A hunter who shot at a lion that chased him hit it"

```
19a) [TL,TH:
        set(indef,1,masc,H,(hunter(H),shot_at(H,TL),
            set(indef,1,neutr,L,(lion(L),chased(L,TH)),TL)),TH),
        hit(TH,TL)]
```

But there are fundamental differences between the treatment of variables in Prolog and DRT: During the interpretation of a Prolog program, bindings of a given variable spread throughout a clause to all occurrences of the same name, forwards and backwards. DRT, however, allows only forwards, "anaphoric", spreading of values. Since a pronoun is processed as soon as it is encountered, it can "look for" antecedents exclusively in the DRSs built up by the *preceding* discourse. The interpretation procedures of DRT thus implement, implicitly, the syntactic rule that a pronoun can refer anaphorically to a *preceding* noun phrase that *c-commands* it. Because this is, at the same time, the only case where anaphora is allowed, these interpretation rules block, correctly, cataphora from the pronoun to the indefinite noun phrase in

23) He said that a boy had taken the book

But legitimate cases of cataphora, such as those in Bach-Peters-sentences, are blocked by these interpretation rules of DRT, as well. Hence we must *weaken* the accessibility restrictions for anaphoric pronominal references somewhat, but not too much: If we modelled them on Prolog's unrestricted variable sharing, 23 would go through in its coreferential reading.

Accessibility rules of DRT not only block certain correct interpretations, they also allow certain blatantly incorrect ones. They would allow, for instance, the sentence above with a *definite* noun phrase, i.e.

24) He said that the boy had taken the book

to get an interpretation where pronoun and definite noun phrases are coreferential. Why? The *correct* interpretation of this sentence (*no* coreference between "he" and "the boy") requires that the definite noun phrase will be able to find an antecedent among the pre-existing discourse referents. But then the sentence-initial "he" would be *equally* capable of accessing them, and this would allow the prohibited coreferential, cataphoric, reading of the "he" (i.e. "pseudo-cataphora" via a common antecedent). The same thing holds for "He hit the lion that chased the hunter who shot at it".

The prohibited reading of this type of sentence can be ruled out on the basis of purely syntactic information. The standard rule about pronouns says that a pronoun cannot be coreferential with an noun phrase if it *both* precedes and c-commands it. This rules out the cataphoric use of a pronoun if it c-commands its target noun phrase but it allows cases of cataphora such as

25) When he got up, John felt hungry

(which are reducible to anaphora) as well as Bach-Peters-sentences (which are not), but it blocks the prohibited coreferences in "He hit the lion that chased the hunter who shot at it" and "He said that the boy had taken the book".

Mittwoch (1983) has shown that these purely syntactic criteria are not sufficiently general to cover all relevant occurrences of cataphora. In many cases, discourse considerations are needed to explain why cataphora is allowed. The pronoun can occur, for instance, in a sentential constituent which is demoted, by explicit discourse subordination markers, to a lower position than warranted by syntax. Thus, in

26) I haven't seen him yet but John is back

(from Mittwoch 1983) the "but" functions as an overt marker of topicality for the second sentence, demoting the first sentence, and in

27) He may not represent the US at the United Nations
anymore, but that does not mean that Andrew Young has
slowed his pace

(from Macleod 1984, quoting from "Time") the "but", together with the modal "may", even manages to make cataphora acceptable from a sentence-initial subject position (at least in journalese). The common element of all these examples of cataphoric pronouns is that they occur in *discourse conditions*. In simple cases this coincides with sentential conditions ("if" etc.), and very often with sub-sentential conditions (in particular with postmodifiers of noun phrases, such as restrictive relative clauses, prepositional phrases, or nonfinite clauses). But the picture is complicated by the fact that the "antecedent" of cataphora must be definite if the sentence is specific. Compare

28) ?? A hunter who shot at it hit a lion that chased him

29) A hunter who shot at it hit the lion that chased him

30) ?? When he was poor a farmer tended to overwork his donkey

31) When he was poor the farmer tended to overwork his donkey

In general (and in many generic) sentences this restriction does not hold. The following sentences are fine although the "antecedent" noun phrases are indefinite:

32) A hunter who shoots at it will hit a lion that chases him

33) If he is poor a farmer will tend to overwork his donkey

What seems to happen here is that, intutively speaking, the cataphoric pronoun sets up an "expectation" for a following noun phrase which is specific or non-specific, depending on the specificity of the conditional context in which the pronoun finds itself. The specificity of the pronominal context is determined (mainly) by the aspect of the verb there: "who shoots" vs. "who shot", "if he is poor" vs. "when he was poor", etc. A specific expectation requires a definite noun phrase or a proper name as its "antecedent", while a non-specific one accepts either an indefinite or a definite noun phrase.

Required Modifications to Discourse Representation Theory

How could DRT incorporate this kind of information in order to determine more reliably the range of permissible anaphora and cataphora, while ruling out the illegal coreferential reading in sentences like "He said that the boy had taken the book"? The following is a list of requirements for an implementation that would take these additional conditions into account: As in traditional DRT, the incoming sentence opens a new DRS, which defines the space where all newly created discourse referents can survive. Noun phrases create set expressions (the "conditions" of standard theory): Indefinite and definite noun phrases give rise to normal set expressions, while pronouns create set expressions of a special type. *Indefinite* noun phrases give rise, in addition, to discourse referents, which are deposited in the DRS under construction. Traditional DRT has proper names create discourse referents, too. Whether this is the best possible decision is open to debate. It would, in many respects, be more consistent to treat proper names on a par with definite noun phrases. Discourse referents should contain all the information that can become relevant for the resolution of pronominial anaphora, i.e. at least number and gender. *Definite* set expressions derived from full noun phrases *without conditional modifiers*, as well as those derived from definite pronouns, are evaluated as soon as they are created, i.e. they try to find their antecedents among the pre-existing discourse referents. Expressions for pronouns whose antecedents have been found are removed, once they have done their duty as value sharing channels. So far nothing really new.

But now the *first* modification of standard theory is needed: Definite *full* noun phrases are not allowed to look for their antecedents inside the DRS still under construction, whereas pronouns may do so. *Second*, when *any* definite noun phrase (full noun phrase *or* pronoun) has found the correct discourse referent, it drags it into the DRS under construction. These two changes make sure that two full noun phrases within the same clause are never interpreted as coreferential. They also block cataphora in "He said that a boy had taken the book" (the "he" has dragged the appropriate discourse referent into the DRS, where it is "invisible" to the subsequent "the boy"). And, finally, it brings a discourse referent accessed by a definite noun phrase into focus and makes it the prime candiate for subsequent anaphoric reference by pronouns. The *third* modification to standard DRT is this: Because pronouns in non-generic contexts require definite noun phrases as antecedents (see examples 28 to 31), discourse referents must also carry information about the definiteness of the noun phrase from which they were derived. Set expressions derived from pronouns will use this information to determine whether a given discourse referent is a possible antecedent. The *fourth* modification, finally, takes care of cataphora: Whenever an expression denoting a *condition* (on the discourse, sentence, or sub-sentential level) is encountered, no embedded DRSs are created (as it is done in standard DRT for "if"- and "every"-sentences) and the production of discourse referents goes on, but evaluation of all new set expressions is suspended. In particular, no further attempts at anaphora resolution are made, and all pronouns encountered from now on are stored in the DRS under construction as unevaluated set expressions. It is only when the end of the clause is reached that unevaluated set expressions are processed. Among the set expressions and discourse referents "in suspended animation" within a DRS, *any* reference (backwards and forwards) is permitted, as long as the conditions outlined above are fulfilled. This allows cataphora to be modelled, while the classical syntactic restrictions (cataphora only from a non c-commanding constituent) are subsumed. Lastly, those discourse referents that are allowed to live (those from assertive, i.e. non-conditional, contexts, and those that were dragged in from the outside) are released into the universe of discourse referents.

ACL 1983. ACL, 1983, *Proceedings of the 21st Annual Meeting of the ACL.*

Brady 1983. Brady, M. and , R.C. Berwick, ed., *Computational Models of Discourse*, (The MIT Press Series in Artificial Intelligence), MIT Press, Cambridge, Mass./London (1983).

Carden 1982. Carden, G., "Backwards anaphora in discourse content," *Journal of Linguistics* **18** pp. 361-387 (1982).

Dik 1973. Dik, S.C., "Crossing Coreference Again," *Foundations of Language* **9** pp. 306-326 (1973).

Frey 1983. Frey, W. and Reyle, U., "Lexical Functional Grammar und Diskursrepräsentationstheorie als Grundlagen eines sprachverarbeitenden Systems," *Linguistische Berichte*, (88) pp. 79-100 (1983).

Groenendijk 1981. Groenendijk, J.A.G., Janssen, T.M.V., and Stok, M.B.J., *Formal Methods in the Study of Natural Language; Part 1*, Mathematisch Centrum Tract 135, Amsterdam (1981).

Guenthner 1983. Guenthner, F. and Lehmann, H., "Rules for Pronominalization," in: Proceedings of the 1st Conference of the European Chapter of the ACL, Association for Computational Linguistics, Pisa (September 1983).

Guenthner 1985. Guenthner, F., "Linguistic Meaning in Discourse Representation Theory," FNS-Bericht 85-5, Forschungsstelle für natürlich-sprachliche Systeme, Universität Tübingen (1985).

Guenthner 1986. Guenthner, F., Lehmann, H., and Schönfeld, W., "A theory for the representation of knowledge," *IBM J Res Develop* **30**(1) pp. 39-56 (1986).

Jacobs 1970. Jacobs, R.A. and , Rosenbaum P.S., eds., *Readings in English transformational grammar.* 1970.

Kamp 1981. Kamp, H., "A Theory of Truth and Semantic Representation," pp. 277-322 in *Groenendijk 1981*, (1981).

Kamp 1983. Kamp, H., *SID without Time or Questions*, Unpublished; strongly rumoured to appear as CSLI Report 1983.

Karttunen 1971. Karttunen, L., "Definite Descriptions with Crossing Coreference," *Foundations of Language* **7** pp. 157-182 (1971).

Kolb 1985. Kolb, H.-P., "Aspekte der Implementation der Diskursrepräsentationstheorie," FNS-Script 85-1, Universität Tübingen, Forschungsstelle für natürlichsprachliche Systeme (May 1985).

Macleod 1984. Macleod, N., "More on backward anaphora and discourse structure," *J of Pragmatics* **8** pp. 321-327 (1984).

McCawley 1970. McCawley, J.D., "Where do noun phrases come from?," pp. 166-183 in *Jacobs 1970*, (1970).

Mittwoch 1983. Mittwoch, A., "Backwards Anaphora and Discourse Structure," *J of Pragmatics* **7** pp. 129-139 (1983).

Pinkal 1986. Pinkal, M., "Situationssemantik und Diskursrepräsentationstheorie; Einordnung und Anwendungsaspekte," pp. 397-407 in *Stoyan 1986*, (1986).

Root 1986. Root, R., "The Semantics of Anaphora in Discourse," Ph.D. Thesis, Universität Tübingen, Forschungsstelle für natürlichsprachliche Systeme, University of Texas at Austin (May 1986).

Stoyan 1986. Stoyan, H. ed., *GWAI 1986; 9th German Workshop on Artificial Intelligence*, (Informatik-Fachberichte), Springer, Berlin etc. (1986).

Webber 1983. Webber, B.L., "So What Can We Talk About Now?," pp. 331-371 in *Brady 1983*, (1983).