

On GB Parsing and Semantic Interpretation

Torbjørn Nordgård
Bergen

Abstract

The paper shows how sentences containing scope ambiguities can be assigned syntactic and semantic structures by means of sloppy deterministic processing techniques only. The semantic framework is Discourse Representation Theory, and the sloppy deterministic parser is described in Nordgård (1993). Of primary concern for the article is the transition from syntactic structures to discourse representation structures (DRSs).

Introduction

In Discourse Representation Theory (DRT) nominal constituents in a syntactic tree are substituted by variables when the tree is translated to a semantic expression which is interpretable wrt. a model, cf. Kamp & Reyle (1992). The variables and the “reduced” trees are crucial parts of *Discourse Representation Structures* (DRSs). Consider example (1) and the DRS (2), which results from (1), assuming that sentence (1) is the first utterance in a context:

(1) Peter likes Mary

(2)

x, y
Peter(x)
Mary(y)
x likes y

The content of the box in (2) constitutes a DRS. The first line is the *variable list*, and the other expressions are the *conditions* of the DRS. The expression *x likes y* is a shorthand for the syntactic representation of (1) where *x* and *y* have replaced **Peter** and **Mary**, e.g. [s x [v_P [v likes] y]]. The variables introduced in a DRS δ have scope over expressions inside δ and all other DRSs “contained” in δ .

Another example is given by (3) and the corresponding DRS in (4)¹.

(3) A man smokes

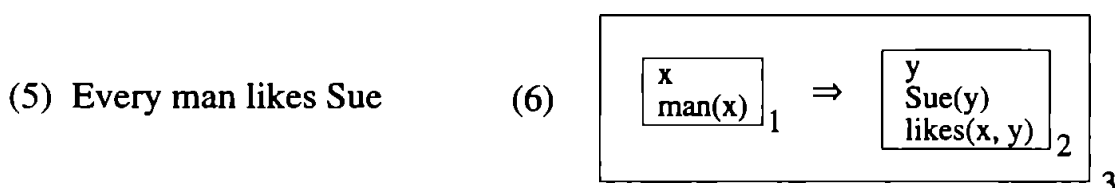
(4)

x
man(x)
smokes(x)

¹The reader is referred to Kamp & Reyle (1992) for elaboration of this analysis of indefinites.

Universal Quantification

Universal quantification is exemplified in (5). According to Kamp & Reyle we want a DRS like (6):



Note that this representation consists of three distinct DRSs, as indicated by the subscripts 1–3. DRS_1 and DRS_2 are subordinate to DRS_3 , which is easily seen from the box notation. One might wonder how condition *likes*(*x*,*y*) in DRS_2 has access to variable *x* in DRS_1 . Kamp & Reyle state that a condition α in some DRS_i has access to variables declared in some DRS_{ii} if DRS_i is subordinate to DRS_{ii} , or DRS_i and DRS_{ii} are connected by “ \Rightarrow ”, DRS_i on the righthand side of “ \Rightarrow ” (this is a simplification of the terms ‘subordinate’ and ‘accessibility’; see Kamp & Reyle (1992) for a detailed exposition).

The translation of universally quantified NPs is performed by a construction rule:

(7)

Triggering

configuration

$[\gamma [\text{NP} [\text{Det every}] [\text{N}' [\text{N } \alpha]]] [\phi \dots]]$ or

$\omega \supseteq \omega', \omega \in \text{CON/DRS}_0: [\phi \text{P} [\phi' [\phi \dots]] [\text{NP} [\text{Det every}] [\text{N}' [\text{N } \alpha]]]]]$

Introduce in CON/DRS₀: New condition $DRS_1 \Rightarrow DRS_2$ where DRS_1 and DRS_2 are empty

Introduce in U/DRS₁: new discourse referent u

Introduce in CON/DRS₁: $\alpha(u)$

Introduce in CON/DRS₂: New condition χ , where χ is the result of substituting u for $[\text{NP} [\text{Det every}] [\text{N}' [\text{N } \alpha]]]$ in ω .

Delete ω from DRS₀.

CON/DRS_n is an abbreviation for the set of conditions in DRS_n , and U/DRS_m is a shorthand for the universe of DRS_m , i.e. the variables declared in DRS_m . We assume that (7) applies as soon as the triggering configuration is detected by the syntactic parser.

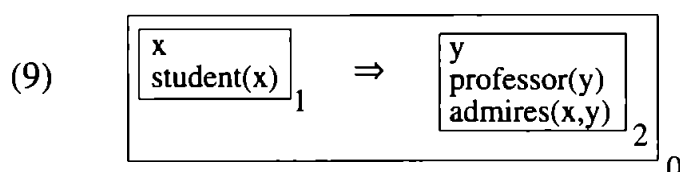
Scope Ambiguities

Sentence (8) is an example of scope ambiguity:

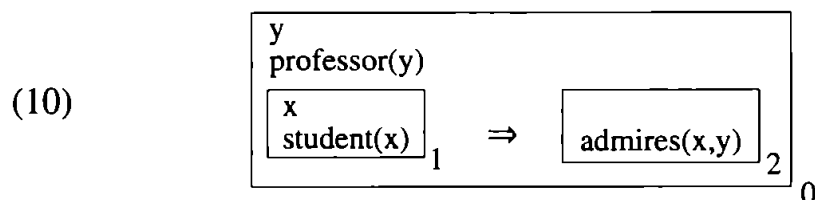
(8) Every student admires a professor

The sentence can either mean that every student admires a particular professor (the wide scope reading of the indefinite phrase), or it can mean that the students admire different professors (the narrow scope reading of the indefinite).

DRT, as presented here and in Kamp & Reyle, assigns a DRS like (9) to this sentence, assuming a top-down left-to-right translation to semantic representations:



The construction rule for indefinites refers to the “current” DRS, and the current DRS is DRS_2 when the translation takes place. Thus, the wide scope interpretation of the existential phrase is lost. This reading should be represented as



Williams' Analysis of Scope Ambiguities

Williams (1986, 1988) proposes a scope theory without quantifier raising in Logical Form. This theory is interesting for the design of natural language processing systems because it avoids operations on phrase structure (LF movements are operations on phrase structure). Williams assumes that “a quantification structure consists of four elements: the quantifier, the variable, the scope and the restriction on quantification” (Williams 1988:136). A restriction is for instance **man** in **every man**; the variable is an empty category or a quantifier *in situ*. In examples like (11) the quantifier is *in situ* and occupies the variable position:

(11) John saw everyone

In sentence (12) the quantifier binds an empty category in the variable position:

(12) What_i did John see e_i

Since a quantifier like **everyone** in (11) doesn't move in Williams's system, its scope must be defined by other means than c-command, which is the standard way of defining the scope of moved quantifiers. Williams assumes that the node *S* (*S*=InflP) restricts the scope of a quantifier.

Consider now the scope analysis of sentence (8) without QR:

(8) [_S Every student [_{VP} admires a professor]]

Node *S* dominates both *every student* and *a professor*.¹ The two quantifiers thus share scope. The scope ambiguity follows straightforwardly if we assume that scope orderings are underdetermined when two or more quantifiers are included in the same scope domain.

Deterministic Processing and Parallel Syntactic and Semantic Structure Building

In this section I will try to show that certain processing techniques and principles developed in Nordgård (1993) are useful in the computation of scope ambiguities in a GB/DRS approach, together with a scope analysis without LF-movements like Williams'. The parser described in Nordgård (1993) is deterministic, sloppy deterministic, to be precise.² It cannot destroy or "forget" structure it has created. Information can, however, be added to its left context, e.g. indices and new constituents. Importantly, such a parser does not waste time on non-well-formed structural representations, and, consequently, it is efficient.³

In the examples below I will assume some familiarity with Nordgård (1993). To recapitulate very briefly, the system has the following important properties: The analysis starts out with a sentential template, e.g. [_{CP} [_{XP}] [_C **X_j**] [_{IP} [_{NP}] [_{I'} e_j] [_{VP} [_{V'} [_v e_j]]]]]]. Positions in boldface, i.e. Spec-CP, Spec-IP, Head-IP and Head-VP, will be considered during the parsing process, and positions without empty

¹Assume, for the moment, that the structure of (8) is [_S [_{every student}] admires [_{a professor}]].

²A sloppy deterministic machine can output a set of analyses for some input string as long as each analysis is computed deterministically. A "standard" deterministic device is only allowed to produce exactly one result.

³If the search space is huge then efficiency decreases, of course. For discussion, see Nordgård (1993), chapter 7.

categories will be instantiated by lexical material. New positions are added on the basis of properties of lexical items introduced into the tree (subcategorized constituents), or non-subcategorized constituents discovered in the input string (adjuncts). The parser's "attention" in the tree is governed by a stack of queues of waiting positions. Positions are represented as integers referring to unique nodes in the tree (cf. (i) below), and they are organized in queues. These queues are in turn organized in a stack. This organization enables the parser to delay parts of the analysis until substructures are analyzed properly. The details are irrelevant in the examples to be discussed below. Finally, the parser makes use of procedural instructions ("heuristic rules") when deciding what to do in a given state (trace insertion, PP attachment, and so on). See Nordgård (1993) for a comprehensive discussion of this parsing system.

In what follows I would like to explore whether DRSs can be built deterministically, and in particular whether scope ambiguities can be captured by deterministic techniques.¹ The most important ideas are as follows:

- DRSs are created in parallel with the syntactic analysis
- Quantifier indices percolate upwards in the tree
- The scope of a quantifier *in situ* is determined by the node where its percolated index is terminated

An Example

Let me show the effects of these ideas by an outline of the syntactic and semantic analysis of sentence (13):

- (13) Jens beundrer enhver professor
Jens admires every professor

Stages in the analysis will be represented as a triple containing the "remaining" string items, the structural representation built "so far", and the DRSs derived from the structural representation "so far":

- (14) a. Input string, b. Tree structure, c. DRS(s)

First the clausal template is initialized (the second line of (i), see below). Each node has a unique identifier (a number attached to the left, e.g. ${}_3C'$) which makes it possible to refer to them in DRSs. Assume that the main DRS is empty in this example:

¹Of course, scope ambiguities must rely on *sloppy* deterministic techniques.

i. *Jens beundrer enhver professor*

[1CP [2XP] [3C' 4X_j [5IP [6NP] [7I' 8e_j [9VP [10V' [11V 12e_j]]]]]]]
 Empty main DRS

The next step is the syntactic analysis of the content of Spec-CP, which turns out to be an NP containing the proper name **Jens**.¹ When the analysis of Spec-CP is completed, information can be put into the DRS. The parser's attention will now be at Head-CP.

ii. *beundrer enhver professor*

[1CP [2NP Jens] [3C' 4X_j [5IP [6NP] [7I' 8e_j [9VP [10V' [11V 12e_j]]]]]]]

$x, \text{Jens}(x), \#1\#, x:\#2\#$

The notation used in the DRS calls for some comments. Numbers enclosed by “#” refer to nodes in the tree. The expression $x:\#2\#$ means that variable x is connected to the position in the tree where node 2 is. If the variable prefix is absent, $\#n\#$ refers to the “current tree”. For the moment this can be taken as simply a notational convenience which replaces the entire tree in Kamp & Reyles notation, but later in this section it will be demonstrated that this notation opens for a flexible account of scope ambiguities.

Next the verb is attached to Head-CP; an empty category is inserted in Spec-IP, and the subcategorized argument of **beundrer** is inserted in the tree. The remaining input string is analyzed as the object of **beundrer**:

iii. \emptyset

[1CP [2NP Jens]_i [3C' 4beundrer_j [5IP e_i [6NP e_i] [7I' 8e_j [9VP [10V' [11V 12e_j [13NP [Det enhver] [N' [N professor]]]]]]]]]]]]

$x, \text{Jens}(x), \#1\#, x:\#2\#$

Construction rule (7) can be applied, and the result is

iv. \emptyset

[1CP [2NP Jens]_i [3C' 4beundrer_j [5IP e_i [6NP e_i] [7I' 8e_j [9VP [10V' [11V 12e_j [13NP enhver professor]]]]]]]]]]

$x, \text{Jens}(x),$	$y, \text{professor}(y)$	\Rightarrow	$\#1\#, x:\#2\#, y:\#13\#$
----------------------	--------------------------	---------------	----------------------------

¹This is not an appropriate occasion for introducing the operations of the parser. The relevant heuristic rules are described in Nordgård (1993).

The DRS in the standard shorthand notation:

(15)

$x, \text{Jens}(x)$
$y, \text{professor}(y)$

 \Rightarrow

$\text{beundrer}(x, y)$

As is well-known, sentences like (13) are ambiguous in a verb second language like Norwegian. In addition to the reading described above it also has an interpretation where **Jens** is the object of **beundrer**. Since the parser assigns both syntactic representations to the string, distinct DRSs will be created:

- v. \emptyset
 [1CP [2NP Jens]_i [3C' 4beundrer]_j [5IP [6[NP [Det enhver]
 [N' [N professor]]] [7I' 8e]_j [9VP [10V' [11V 12e]_j [13NP e_i]]]]]]]]]

$x, \text{Jens}(x)$
$y, \text{professor}(y)$

 \Rightarrow

$\#1\#, x:\#2\#, y:\#6\#$

In standard notation:

(16)

$x, \text{Jens}(x)$
$y, \text{professor}(y)$

 \Rightarrow

$\text{beundrer}(y, x)$

Hence, the system does output the desired set of DRSs when input sentences are structurally ambiguous.

Scope Ambiguities and Index Percolation

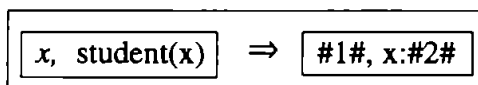
Let us now turn to scope ambiguities:

- (17) *Enhver student liker en professor*
every student likes a professor

As in the previous example, this clause is structurally ambiguous. Space considerations do not permit a discussion of both syntactic readings and their semantic implications. We will consider only the reading where *enhver student* is the subject.

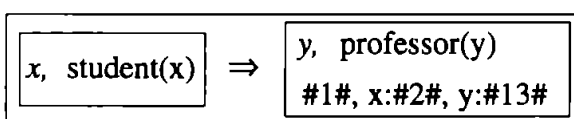
When *enhver student* has been attached to Spec-CP, the main DRS, assumed to be initially empty in this example, is to be modified:

- (18) *liker en professor*
 [1CP [2NP enhver student] [3C' 4X_j [5IP [6NP] [7I' 8e_j
 [9VP [10V' [11V 12e_j]]]]]]]]



Attachment of the verb and projection of the object position are as in the previous example. When the object is syntactically analyzed, the narrow scope reading of the existential phrase is obtained:

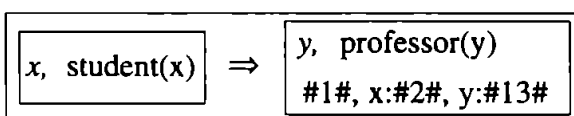
- (19) ∅
 [1CP [2NP enhver student]_i [3C' 4liker_j [5IP [6NP e_i] [7I' 8e_j
 [9VP [10V' [11V 12e_j] [13NP [Det en] [N' [N professor]]]]]]]]]]]



If the DRS in (18) is the only one available, the wide scope reading of the object is lost. To obtain both interpretations *index percolation* comes into play. Recall from Williams' system that scope ambiguity arises when two quantifiers share scope domain. The idea to be elaborated here is that scope ambiguity arises as soon as an index percolates to a position in the tree which dominates (or equals) another quantifier. If the index of the object percolates to CP in (19), the indefinite takes scope over the subject.¹

Suppose that the index of quantified phrases can percolate up to any maximal projection whose head has semantic content, where 'semantic content' is to be understood as 'ability to assign thematic roles'. Note that the syntactic analysis assumed here implies that the index can percolate to CP in root clauses because the verb moves to Head-CP. Thus, the index of the existential phrase will at least percolate to VP, but since VP does not dominate node #2 no new interpretation can be derived. When index #13 reaches CP the following configuration (20) results:

- (20) [1CP:13 [2NP enhver student]_i [3C' 4liker_j [5IP [6NP e_i] [7I' 8e_j
 [9VP [10V' [11V 12e_j] [13NP [Det en] [N' [N professor]]]]]]]]]]]



¹A possible alternative is that scope ambiguity arises as soon as some percolating index dominates an EC bound by another quantifier. If so, percolation to IP is sufficient in the example under consideration.

Now the index of the object is connected to a position (node #1) which dominates the position held by the other quantifier, i.e. node #2. If a DRS is created in this configuration another scope interpretation results provided that the constituent referred to by index #13 is translated first:

(21) $\boxed{y, \text{professor}(y)}$
 $\boxed{\#1\#, y:\#13\#}$

In this representation the variable y has scope over the entire clause, assuming that its structural representation is the same:

(22) $\boxed{y, \text{professor}(y)}$
 $\boxed{x, \text{student}(x)} \Rightarrow \boxed{\#1\#, x:\#2\#, y:\#13\#}$

Even though we are talking about distinct DRSs representing different scope orderings, the DRSs share some information. For DRS (20) and DRS (22) the global DRS existing prior to the analysis of the clause is common. By assuming an empty discourse in the examples under discussion this point is perhaps not so obvious, but it must nevertheless be taken into consideration. Assume that each node in the tree has a corresponding DRS. We need not be concerned about how the correspondence is made technically, but the corresponding DRS should be a copy of the “current” DRS prior to the analysis of the daughters of the relevant node. Thus, the corresponding DRS of node #1 in (20) is a vacuous structure because the initial DRS was assumed to be empty. Given these assumptions, consider state (ii) from the processing example above:

ii. *beundrer enhver professor*

[₁CP [₂NP Jens] [₃C' ₄X_j [₅IP [₆NP] [₇T' ₈e_j [₉VP [₁₀V' [₁₁V ₁₂e_j]]]]]]]]

$\boxed{x, \text{Jens}(x), \#1\#, x:\#2\#}$

₁CP: Empty DRS

₂NP: Empty DRS

₃C': $\boxed{x, \text{Jens}(x), \#1\#, x:\#2\#}$

Information is not put into any DRS until a node yielding such information is analyzed. Note, in particular, that the DRS connected to a node which “triggered” some information is not affected by “its own semantic information”. That is, the DRS connected to node 2 is not modified by the semantic content extracted from node 2. This information is passed onto node 3. The last DRS created in an analysis is one of possibly more final results.

Given these assumptions, it is fairly straightforward to build alternative DRSs for sentences with scope ambiguities: If the index percolation process shows that a percolated index i of some quantifier Q is attached to some node X which dominates another quantifier P , a new DRS can be made as a corresponding DRS of X . In the new DRS the variables introduced by Q are introduced. To preserve determinism, the index percolation process must take place deterministically. If we adhere to the tree searching strategy developed in Nordgård (1993), the process can informally proceed as follows: Whenever a relevant quantified expression Q is detected, check whether there is a c-commanding quantifier P higher up in the tree.¹ If so, put an index of Q on the maximal projection MP dominating P . Create a new DRS based on the DRS connected to MP . Introduce the variables introduced by Q here. Restart the analysis from this point.²

Applied to the example above, the processing starts up with node 1CP again, but now the corresponding DRS contains the information in (21). Provided that the same variable is not introduced again when node 13 is translated, the wide scope analysis of the object phrase is achieved.

Conclusion

This paper has demonstrated that the processing system developed in Nordgård (1993) can be related to DRT in a way which preserves the deterministic nature of the syntactic parser. In particular, scope ambiguities can be handled by deterministic techniques. I believe this is an important result because it shows that scope ambiguities do not enforce guessing algorithms.

¹This search can be accomplished by deterministic finite state machinery, cf. Nordgård (1993, chapter 7) for discussion.

²This strategy presupposes that a copy of the remaining string elements is stored together with the DRSs of the nodes. One might object that it seems unnecessary to perform the analysis once more. This is presumably true.

References

- Kamp, Hans and Uwe Reyle. 1992. *From Discourse to Logic*. Ms, University of Stuttgart.
- Nordgård, Torbjørn. 1993. *A GB-Related Parser for Norwegian*. Peter Lang Inc., Bern.
- Nordgård, Torbjørn. 1992. *On Existential NPs, Multiple Interrogation and Scope*. Unpublished ms., Department of Linguistics and Phonetics, University of Bergen.
- Williams, Edwin. 1986. *A Reassignment of the Functions of LF*. In LINGUISTIC INQUIRY 17.2.
- Williams, Edwin. 1988. *Is LF Distinct from S-Structure? A Reply to May*. In LINGUISTIC INQUIRY 19.1.