# On the Complexity of Queries for Structurally Annotated Linguistic Data

Laura Kallmeyer
SFB 441 “Linguistic Data Structures”
University of Tübingen
Köstlinstr. 6, 72074 Tübingen, Germany
lk@sfs.nphil.uni-tuebingen.de

## Abstract

*This paper addresses on a theoretical level the problem of extracting information out of syntactically annotated corpora. More precisely it is concerned with the question of the descriptive complexity of query languages for structurally annotated data. This can be divided into two aspects: On the one hand, the demands linguists make on query languages need to be specified and on the other hand, the queries that are relevant from a linguistic point of view must be examined with respect to the logical expressive power needed to formulate them.*

*The paper does not provide general answers to these questions. It rather examines two concrete examples, namely scrambling phenomena in German annotated in the NEGRA annotation style and long-distance wh-movement in the Penn Treebank. Using Ehrenfeucht-Fraïssé games, it is shown that one needs at least three variables to express the corresponding relevant queries in first order logic.*

*The paper presents a first step towards a more systematic theoretic evaluation of query languages for structurally annotated corpora.*

## 1. Introduction

The work presented here is motivated by two demands in the area of query languages for syntactically annotated corpora. First, although there are several query languages for corpora, such as xkwic ([3], [4]) and tgrep ([8]), theoretical foundations for research on queries for annotated corpora are missing. The second problem is that existing query-languages like tgrep for the Penn Treebank (bracketing style I) or query languages for SGML-annotated corpora are not sufficient because they assume that the annotations are finite trees. However, in corpora such as the NEGRA corpus developed in Saarbrücken ([10]) or the Penn Treebank II edition annotated at the University of Pennsylvania ([7], [1]), this is no longer the case. In the NEGRA corpus crossing branches are allowed and in the Penn Treebank II there is a binary coindexation relation besides the relations immediate dominance, dominance and linear precedence.

This paper approaches on a theoretical level the question of the descriptive complexity of query languages for structurally annotated data. A query is a (partial) description of structures and it can be expressed in an adequate logic. When applying the query to a corpus, the structures that satisfy the corresponding logical expression (i.e. its models) are extracted from the corpus. In contrast to constraint based grammar formalisms, in the case of query languages, it is not necessary to express general principles and consequently one can choose a rather simple logic. With fragments of first order logic over finite graphs one can probably already express most of the things needed by linguists. The interesting question is how small this fragment might be. This can be divided into two aspects:

1. Which kind of queries are linguistically relevant?
2. How much expressive power is needed to formulate these queries?

I will not give general answers to these questions in this paper but I will rather look at two concrete examples namely the recognition of scrambling constructions in German and long-distance wh-movement in English. For these two phenomena, possible queries are considered and using techniques from finite model theory, in particular Ehrenfeucht-Fraïssé games, the minimal amount of variables needed to express these queries in first order logic is determined. Since the number of variables of a query is crucial for the complexity of the query (see [6]) the specification of the minimal number of variables needed to express certain queries is an important issue that is not only of theoretical but also of practical interest.

The structure of the paper is as follows: Section 2 presents two phenomena linguistis might want to search for in structurally annotated corpora, and corresponding queries are proposed. In Section 3, these queries are examined more closely, and the maximal number of variables needed to express them in first-order logic is investigated.

## 2. Linguistically relevant queries

In this section, I will consider two possible queries for syntactically annotated corpora: the search for scrambling construction in the NEGRA corpus and queries for long-distance wh-movement in the Penn Treebank II. The reason why these two queries are chosen is that in both cases, one is looking for non-local phenomena in a certain sense. Therefore the query must express relations between parts of the annotation that are not directly connected. Because of this, such phenomena are promising candidates for rather complex queries.

I have chosen the Penn Treebank II and the NEGRA corpus, since these are, as far as I know, at least for English and German the only treebanks with an annotation that is rich enough to contain predicate-argument relations. The annotation in the German Verbmobil corpora ([11]) for example is a rather flat annotation based on the theory of topological fields that does not encode predicate-argument relations.

### 2.1. Scrambling in the NEGRA corpus

Scrambling in the German *mittelfeld* is interesting for linguists for various reasons. It is for example important for linguists interested in German modal verbs because it serves to recognize coherent constructions. Also, it is interesting in general that one semantic relation can be expressed by different syntactic variants. Therefore queries for scrambling configurations must be possible, provided that the annotation is such that these constructions can be found.

As an example for scrambling consider (1)b. (in contrast to (1)a. where scrambling did not take place).

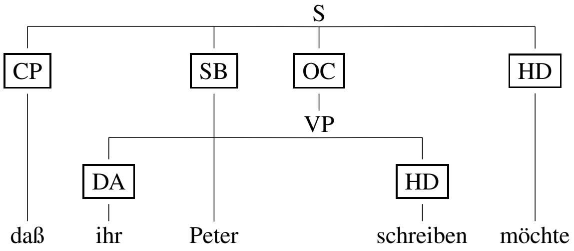
(1) a. ... daß Peter ihr schreiben möchte
... that $\text{Peter}_{\text{nom}}$ $\text{her}_{\text{dat}}$ write wants
... that Peter wants to write her

b. ... daß ihr Peter schreiben möchte
... that $\text{her}_{\text{dat}}$ $\text{Peter}_{\text{nom}}$ write wants
... that Peter wants to write her

The pronoun *ihr* is a dative object of *schreiben*. In the unmarked case (no scrambling) in (1)a., it appears next to the verb *schreiben*, whereas in the scrambling case (1)b. it is moved to the left of the subject of the matrix clause.

In the NEGRA corpus developed in Saarbrücken (see [10]) the difference between the corresponding structures is expressed by the syntactic annotation since predicate argument structures are encoded in the NEGRA annotation. The annotation of (1)b. shown in (2), contains crossing edges because *ihr* is scrambled in front of *Peter*. The edge labels are CP for compelementizer, DA for dative, SB for subject, OC for clausal object and HD for head.

(2)




When designing a query for these kinds of scrambling configurations, one must make sure that also the annotations of the following sentences satisfy the query:

(3) a. ... daß ihr Peter möchte schreiben
... that $\text{her}_{\text{dat}}$ $\text{Peter}_{\text{nom}}$ wants write
... that Peter wants to write her

b. ... daß ihr mein Bruder schreiben möchte
... that $\text{her}_{\text{dat}}$ my $\text{brother}_{\text{nom}}$ write wants
... that my brother wants to write her

c. ... daß den Brief Peter schreiben möchte
... that the $\text{letter}_{\text{acc}}$ $\text{Peter}_{\text{nom}}$ write wants
... that Peter wants to write the letter

A possible first order query would be:

There is a node $x_1$ with label S that dominates a $x_2$ and has a complement clause $x_3$. $x_2$ is not the head of $x_1$, and $x_3$ has the label VP. $x_3$ dominates a node $x_4$ and it has a head $x_5$, such that $x_4$ is left of $x_2$ and $x_2$ is left of $x_5$.

Here "is left" means either immediate linear precedence or its transitive closure.

(4)
$$\begin{array}{l}\exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \\ \quad (S(x_1) \wedge x_1 \lhd^+ x_2 \wedge \neg HD(x_1, x_2) \wedge \\ \quad VP(x_3) \wedge x_3 \lhd^+ x_4 \wedge HD(x_3, x_5) \wedge \\ \quad x_4 \prec^+ x_2 \wedge x_2 \prec^+ x_5)\end{array}$$

$\lhd$ stands for the parent (immediate dominance) relation, and $\prec$ stands for immediate linear precedence between terminals (not between internal nodes). The superscript $+$ indicates the transitive closure of a relation.

### 2.2. Wh-movement in the Penn Treebank

Another interesting phenomenon that linguists might want to search in a corpus are long-distance wh-movements as in (5) (from [5]).

(5) a. $\text{Whom}_i$ do you think that Lord Emsworth will invite $t_i$?

b. $\text{Whom}_i$ do you think Lord Emsworth will invite $t_i$?

c. $^*\text{Who}_i$ do you think that $t_i$ will invite Poirot?

d. Who$_i$ do you think $t_i$ will invite Poirot?

In (5)a. for example *whom* is an argument of *invite*, i.e. of the embedded clause. In a Government and Binding analysis as in [5], the wh-element *whom* is assumed to have moved from the embedded clause to the top of the structure.

For a linguist such constructions are interesting for several reasons. It is for example an interesting question whether evidences for principles like the Empty Category Principle (ECP) that is supposed to be responsible for the ungrammaticality of (5)c. can be found in large corpora or whether one can even find violations of such principles. Therefore a linguistically adequate query language should allow the search for constructions as in (5).

In order to find such constructions, one needs a corpus with traces or with explicit argument structure. The Penn Treebank with the Treebank II style bracketing ([7], [1]) contains traces and consequently allows to search for wh-movements. (6) shows the annotation of (5)a. in the Penn Treebank II bracketing style (there are no intermediate traces in the Penn Treebank).

(6)
```
(SBARQ (WHNP-1 whom)
  (SQ do
    (NP-SBJ you)
    (VP think
     (SBAR that
       (S (NP-SBJ Lord E.)
          (VP will
           (VP invite
             (NP *T*-1))))))))
```

In order to search for long-distance wh-movements as in (5), one would have to formulate the following query:

There are nodes $x_1$ and $x_2$ such that $x_1$ has a label `WHNP` and $x_2$ has a label `*T*` (i.e. is a trace) and $x_1$ and $x_2$ have the same index. Furthermore, there is an `SBARQ`-node $x_3$ immediately dominating $x_1$ such that between $x_3$ and $x_2$, there is (at least) one node $x_4$ with label `SBAR`.

If we supppose that coindexation is denoted by a binary relation symbol $ind$, this query can be expressed in first order logic as follows:

$$(7)\quad \begin{array}{l} \exists x_1 \exists x_2 \exists x_3 \exists x_4 \\ \quad (\mathtt{WHNP}(x_1) \wedge \mathtt{*T*}(x_2) \wedge ind(x_1, x_2) \wedge \\ \quad \mathtt{SBARQ}(x_3) \wedge x_3 \lhd x_1 \wedge \mathtt{SBAR}(x_4) \wedge \\ \quad x_3 \lhd^* x_4 \wedge x_4 \lhd^* x_2) \end{array}$$

## 3. How many variables are necessary?

One problem with the queries in (4) and (7) is that we need more relations than the relations that are directly available in the annotation. An annotation is a graph with the two relations immediate dominance ($\lhd$) and immediate linear precedence ($\prec$). However, as I have shown above, the transitive closures of the relations $\lhd$ and $\prec$ are also needed for linguistic queries. Therefore we have to define transitive closure, and this means (cf. [6]) that first order logic is not enough. We have to add least fixed point operators in order to formulate inductive definitions. The definition of $\prec^+$ and $\lhd^+$ using fixpoint operators is as follows.

$$(8)\quad \begin{array}{lcl} x \prec^+ y & \equiv & x \prec y \vee \exists z (z \prec y \wedge x \prec^+ z) \\ x \lhd^+ y & \equiv & x \lhd y \vee \exists z (z \lhd y \wedge x \lhd^+ z) \end{array}$$

In general, the transitive closure of a $k$-ary relation can be computed in at most $n^k$ steps where $n$ is the size of the graph. In the special case of $\prec^+$ and $\lhd^+$, the relations are computed in at most $n - 1$ steps.

The query in (4) contains five variables. Since the amount of variables needed to express a query is crucial for the complexity of the query (see [6]), it is interesting to look for alternative formulations with less than 5 variables. Leaving the fixed point operators aside and treating $\prec^+$ and $\lhd^+$ as already defined (i.e. as precompiled relations in the graph), the minimal number of variables that is still needed is important for complexity considerations.

An alternative formula with only 3 different variables expressing the same query as (4) is (9):

$$(9)\quad \begin{array}{ll} \exists x \exists y \, [ & [\exists z (S(z) \wedge z \lhd^+ x \neg HD(z, x) \wedge VP(y))] \\ & \wedge \\ & [\exists z (y \lhd^+ z \wedge z \prec^+ x))] \\ & \wedge \\ & [\exists z (y \lhd z \wedge x \prec^+ z)]\, ] \end{array}$$

The second query that was considered above, (7), contains four variables. For this query, an alternative formulation with three variables is also possible:

$$(10)\quad \begin{array}{lll} \exists x \exists y \, [ & [\exists z & (\mathtt{WHNP}(z) \wedge \mathtt{SBARQ}(y) \wedge \\ & & \mathtt{*T*}(x) \wedge y \lhd^* z \wedge ind(z, x))] \\ & \wedge & \\ & [\exists z & (\mathtt{SBAR}(z) \wedge y \lhd^* z \wedge z \lhd^* x)]\, ] \end{array}$$

In the following I will show that for both queries considered here 3 variables is in fact the minimum. The query in (4) and (9) distinguishing between (1)a. and (1)b. cannot be expressed with less than 3 variables, and the same holds for (7) and (10). I will show this using Ehrenfeucht-Fraïssé games.

### 3.1. Ehrenfeucht-Fraïssé games

Ehrenfeucht-Fraïssé games are a logical technique used to show results about the expressivity of certain logics. The idea is the following: Suppose that one wants to know whether a certain query is expressable in a certain logic.

Then one picks two structures over the same signatures such that one satisfies the query and the other does not. Now a game is played by two players. The players mark some of the elements of the structures by putting pebbles on them. One of the players tries to point out the difference between the two structures and the other tries to keep the structures isomorphic. The first player is called the spoiler and the second the duplicator. When the spoiler puts a pebble on an element, this corresponds to an existential quantification. If the pebble has not been used before, this is similar to using a new variable. If it has been used before, then this is the same as reusing a variable. The spoiler tries to move his pebbles in such a way that the substructures marked by the pebbles in the two graphs become different. Whereas the duplicator always tries to answer in such a way that these substructures are isomorphic.

**Definition 1 (Ehrenfeucht-Fraïssé game)** *An Ehrenfeucht-Fraïssé game is played on two structures $G$ and $H$ by two players $S$ and $D$ using $k$ pairs of pebbles. In each move, $S$ (the spoiler) chooses one pair of pebbles. He puts one of the two pebbles on an element in one of the two structures. $D$ (the duplicator) then responds by placing the other pebble of the pair on an element of the other structure. $D$ wins a round of the game iff the two substructures of $G$ and $H$ induced by the set of elements that are marked by the pebbles are isomorphic. $D$ wins the game iff he wins every single round.*

If $D$ has a winning strategy in an Ehrenfeucht-Fraïssé game with $k$ pairs of pebbles on some structures $G$ and $H$, then this means that the difference between $G$ and $H$ cannot be expressed using only $k$ variables.

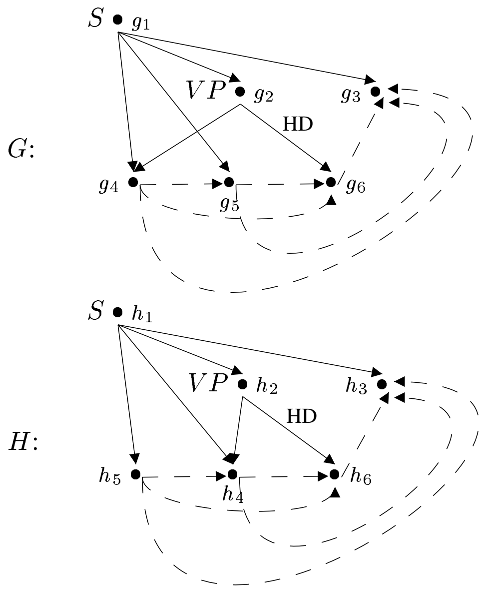
### 3.2. Scrambling needs at least three variables

**Proposition 1** *In order to express scrambling, i.e. the query in (9) in first order logic over structures with dominance, linear precedence and node and edge labels, at least three different variables are needed.*

I will show this playing an Ehrenfeucht-Fraïssé game with two pebbles on graphs that differ with respect to this query.

(11) shows two graphs $G$ and $H$. The arrows represent the dominance relation between the nodes. The dashed arrows represent linear precedence. I am using arrows for dominance instead of simple lines as it is usually done in order to make clear that these are directed edges. $G$ satisfies the query that distinguishes scrambling from non-scrambling cases and $H$ does not satisfy this query.

(11)




In a game with 3 pairs of pebbles on $G$ and $H$, $S$ has a winning strategy that is reflected by (9): First he chooses $g_5$ (corresponding the $\exists x$ in (9)) and in the next step he puts another pebble on $g_2$. Then the third pebble is put on $g_1$, in the next round it is moved from $g_1$ to $g_6$ and in the following round from $g_6$ to $g_4$. As can be easily checked, if $D$ has not lost yet, he looses in this last round.

**Proof of Prop. 1** In order to show that two variables are not enough to express the difference between $G$ and $H$, I will play the Ehrenfeucht-Fraïssé game with two pebbles on the structures $G$ and $H$. In this game, the duplicator $D$, who wants to keep the parts marked by the pebbles isomorphic, has a winning strategy.

Consider substructures of $G$ and $H$ containing two elements. It is easy to see that the following holds: For all $1 \leq i < j \leq 6$, $\langle \{g_i, g_j\}, S, VP, HD, \lhd^+, \prec^+ \rangle$ is isomorphic to $\langle \{h_i, h_j\}, S, VP, HD, \lhd^+, \prec^+ \rangle$. Therefore, $D$ has a simple winning strategy that keeps the marked substructures always isomorphic:

If $S$ chooses a pair of pebbles and puts one pebble on the $i$th element of one structure ($1 \leq i \leq 6$), then $D$ puts the other pebble on the $i$th element of the other structure.

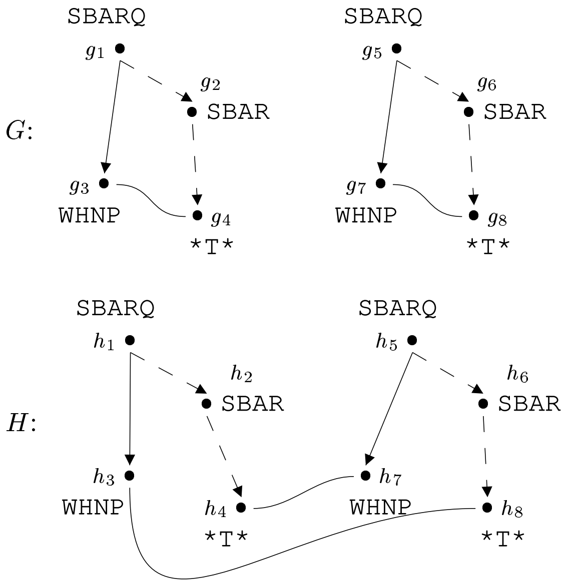
$\square$

### 3.3. Three variables as minimal for wh-movement

**Proposition 2** *In order to express the query (10) in first order logic over structures with the relations immediate dominance, dominance and coindexation and node and edge labels, at least three different variables are needed.*

As in the scrambling case, I will show this by playing an Ehrenfeucht-Fraïssé game with two pebbles on graphs that differ with respect to this query.

(12)




In the graphs in (12), the directed edges (lines) stand for immediate dominance, the directed dashed edges are dominance relations, and the undirected edge stands for the (reflexive) coindexation. The first graph $G$ satisfies the query (10) wheras the second graph $H$ does not satisfy it.

In an Ehrenfeucht-Fraïssé game with three pairs of pebbles, the spoiler has a winning strategy, namely the one reflected by (10): First he puts pebble one on $g_4$, then pebble two on $g_1$, then pebble three on $g_3$, and finally he moves pebble three to $g_2$. For the first three moves, the duplicator must answer in such a way that either $h_4, h_5$ and $h_7$ (in this order) are marked or $h_8, h_1$ and $h_3$. In both cases the duplicator looses in the next move.

**Proof of Prop. 2** In order to show that two variables are not enough to express the difference between $G$ and $H$ in (12), I will play the Ehrenfeucht-Fraïssé game with two pebbles on the structures $G$ and $H$. In this game, the duplicator $D$, who wants to keep the parts marked by the pebbles isomorphic, has a winning strategy.

Consider substructures of $G$ and $H$ containing two elements. These substructures can be divided into equivalence classes (with respect to labels and binary relations immediate dominance, dominance and coindexation). Suppose that the elements of these substructures are orderd.

Notation: $S(g_i, g_j)$ stands for the substructure with $g_i$ as first element and $g_j$ as second element, $S(h_i, h_j)$ stands for the substructure with $h_i$ as first element and $h_j$ as second element for $1 \leq i, j \leq 8$ and $i \neq j$.

The equivalence classes of two-elementary substructures of $G$ and $H$ are:

1. $\{S(g_1, g_2), S(g_5, g_6), S(h_1, h_2), S(h_5, h_6)\}$
2. $\{S(g_1, g_3), S(g_5, g_7), S(h_1, h_3), S(h_5, h_7)\}$
3. $\{S(g_1, g_4), S(g_5, g_8), S(g_1, g_8), S(g_5, g_4),$ $S(h_1, h_4), S(h_5, h_8), S(h_1, h_8), S(h_5, h_4)\}$
4. $\{S(g_2, g_3), S(g_2, g_7), S(g_6, g_3), S(g_6, g_7),$ $S(h_2, h_3), S(h_2, h_7), S(h_6, h_3), S(h_6, h_7)\}$
5. $\{S(g_2, g_4), S(g_6, g_8), S(h_2, h_4), S(h_6, h_8)\}$
6. $\{S(g_3, g_4), S(g_7, g_8), S(h_3, h_8), S(h_7, h_4)\}$
7. $\{S(g_1, g_6), S(g_5, g_2), S(h_1, h_6), S(h_5, h_2)\}$
8. $\{S(g_2, g_8), S(g_6, g_4), S(h_2, h_8), S(h_6, h_4)\}$
9. $\{S(g_1, g_7), S(g_5, g_3), S(h_1, h_7), S(h_5, h_3)\}$
10. $\{S(g_3, g_8), S(g_7, g_4), S(h_3, h_4), S(h_7, h_8)\}$
11. $\{S(g_1, g_5), S(g_5, g_1), S(h_1, h_5), S(h_5, h_1)\}$
12. $\{S(g_2, g_6), S(g_6, g_2), S(h_2, h_6), S(h_6, h_2)\}$
13. $\{S(g_3, g_7), S(g_7, g_3), S(h_3, h_7), S(h_7, h_3)\}$
14. $\{S(g_4, g_8), S(g_8, g_4), S(h_4, h_8), S(h_8, h_4)\}$

Now the strategy of the duplicator in the 2-pebble Ehrenfeucht-Fraïssé game is as follows:

In the first move, if the spoiler puts a pebble on the $i$-th element of one of the structures ($1 \leq i \leq 8$), then the duplicator answers with putting a pebble on the $i$-th element of the other structure.

In the second move, wherever the spoiler puts the next pebble, the duplicator answers in such a way that

(a) the two structures are isomorphic, and

(b) the elements marked in the first move are either both the first elements of the two structures or they are both the second elements of the two structures.

(There are sometimes more than one possible answers.) This is possible because for all equivalence classes $C$, all $i$ with $1 \leq i \leq 8$ and all $k$, $1 \leq k \leq 2$: there is a structure in $C$ with $g_i$ as $k$-th element iff there is a structure in $C$ with $h_i$ as $k$-th element.

In every further move the duplicator can always answer in such a way that (a) and (b) still hold. This is the case because for all equivalence classes $C_1, C_2$ and for $k$, $1 \leq k \leq 2$: If one of the structures in $C_1$ can be transformed to a structure in $C_2$ by changing the $k$-th element, then all structures in $C_1$ can be transformed to a structure in $C_2$ by changing the $k$-th element.

□

This shows that, even if we suppose that $\prec^+$ and $\lhd^+$ are already defined and therefore can be considered as part of the structures, we still need three variables to express the queries for scrambling and wh-movement.

Another interesting aspect also related to complexity considerations is the number of quantifiers that is needed. We have shown that one needs three variables to express the queries in (9) and (10). In (9), we use five quantifiers wheras in (10), four quantifies are needed. In general, it is probably impossible to give an upper bound of the number of quantifiers needed for linguistically interesting queries. The reason is the following: The number of quantifiers roughly corresponds to the number of elements the query refers to. Clearly, for two different queries, it should always be possible to consider only those structures that satisfy both queries at the same time. But this means building the conjunction and therefore adding the numbers of quantifiers of the two single queries.

## 4. Conclusion and further research

This paper investigates the descriptive complexity of queries for syntactically annotated corpora. As examples of concrete queries that are linguistically relevant, I have considered the description of German scrambling constructions assuming that one is dealing with a corpus that is annotated in a way similar to the NEGRA corpus in Saarbrücken and the search for long-distance wh-movement in the Penn Treebank II annotation. Using Ehrenfeucht-Fraïssé games I have shown that in both cases one needs at least three different variables in order to express this query in first order logic over finite graphs, even if linear precedence $\prec^+$ and dominance $\lhd^+$ are considered as part of the graphs, i.e. as already defined.

The overall approach taken in this paper namely the specification of the descriptive complexity of certain queries and of query languages in general under a logical perspecitve has been very little considered so far. However, with the increasing availability of large amounts of syntactically annotated text corpora, the theoretical issues involving query languages for these corpora become more and more important. In a sense this paper presents a first step towards a more systematic theoretical evaluation of query languages.

One problem this paper does not deal with is the relation between the complexity of the query language one needs and the properties of the annotation that are detemined by the annotation scheme. The specific annotation style already restricts the set of possible models for the queries by specifying certain properties the models must have. It is therefore not necessary to express these properties in the query. This has consequences for the logic needed as query language.

Another aspect is the search for more abstract linguistic phenomena. For somebody interested in generative syntax it might for example be interesting to search for syntactic elements responsible for locality phenomena in movement operations (barriers in the sense of [2] or intervening elements in the sense of [9]). This last aspect was suggested to me by Aria Adli, and we are planning to work on these problems in the future. The question is whether annotations of existing corpora allow at all to search for such things, and if the annotations are rich enough, then one might investigate the complexity of the corresponding queries.

## Acknowledgments

The work presented here is part of a project in the SFB 441 "Linguistic Data Structures" funded by DFG (Deutsche Forschungsgemeinschaft). I would like to thank Aria Adli and Reimar Müller, who are both working in other projects of SFB 441, for their valuable comments on the linguistic relevance of queries like those examined in this paper. Furthermore I would like to thank Ann Bies at the University of Pennsylvania who kindly explained some details of the Penn Treebank annotation to me.

## References

[1] A. Bies, M. Ferguson, K. Katz, and R. MacIntyre. Bracketing Guidelines for Treebank II Style Penn Treebank Project. University of Pennsylvania, 1995.

[2] N. Chomsky. *Barriers*. MIT Press, Cambridge, Mass., 1986.

[3] O. Christ. A Modular and Flexible Architecture for an Integrated Corpus Query System. In *Proceedings of COMPLEX'94*, 1994.

[4] O. Christ and B. M. Schulze. Ein flexibles und modulares Anfragesystem für Textkorpora. In H. Feldweg and E. W. Hinrichs, editors, *Lexikon und Text. Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen*, pages 121 – 133. Niemeyer, Tübingen, 1996.

[5] L. Haegeman. *Introduction to Government and Binding Theory*. Blackwell, Oxford UK, Cambridge USA, 2nd edition, 1994.

[6] N. Immerman. *Descriptive Complexity*. Graduate texts in computer science. Springer, New York, 1999.

[7] M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The Penn Treebank: Annotating Predicate Argument Structure. In *ARPA '94*, 1994.

[8] R. Pito. *tgrepdoc – documentation for tgrep*, 1993.

[9] L. Rizzi. Relativized minimality effects. March 1998.

[10] W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. An Annotation Scheme for Free Word Order Languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C., 1997.

[11] R. Stegmann, H. Schulz, and E. W. Hinrichs. Stylebook for the German Treebank in VERBMOBIL. Tübingen, July 1998.