AN INDEXING TECHNIQUE FOR IMPLEMENTING COMMAND RELATIONS

Longin Latecki

University of Hamburg
Department of Computer Science
Bodenstedtstr. 16, 2000 Hamburg 50
Germany

TOPIC AREA: SYNTAX, SEMANTICS

ABSTRACT

Command relations are important tools in linguistics, especially in anaphora theory. In this paper I present an indexing technique which allows us to implement a simple and efficient check for most cases of command relations which have been presented in linguistic literature. I also show a wide perspective of applications for the indexing technique in the implementation of other linguistic phenomena in syntax as well as in semantics.

0. INTRODUCTION

Barker and Pullum (1990) have given a general definition of command relations. Their definition covers most cases of command relations that have been presented in linguistic literature. I will present here an indexing technique for syntax trees which allows us to implement a check for all command relations which fulfill the definition from Barker/Pullum (1990). The indexing technique can be implemented in a simple and efficient way without any special requierments for the formalism used. Hence, the indexing technique has a wide spectrum of applications for testing command relations in syntactic analysis. Futhermore, this method can also be used for command tests in semantics, i.e. to test for any two semantic representations whether the corresponding nodes of the syntax tree are in a command relation. The usefulness and necessity of a command test in semantics have been demonstrated in Latecki/Pinkal (1990).

The general idea of the indexing technique is the following: while a syntax tree is being built, special indices are assigned to the nodes of this tree. Afterwards we can check whether a command relation between two nodes of this tree holds by merely examining simple set-theoretical relations of corresponding index sets.

1. A GENERAL DEFINITION FOR COMMAND RELATIONS

The general command definition from Barker/Pullum (1990) can be informally stated in the following way:

1.1 DEFINITION (all). α P-commands β iff every node with a property P that properly dominates α also dominates β.

In this chapter I will show that this definition is equivalent to the following definition (minimum).

1.2 DEFINITION (minimum). α P-commands β iff the first node with a property P that properly dominates α also dominates β.

In this definition the first node that dominates α means the node most immediately dominating α, as it is usually used in linguistics. Below I will specify both of these definitions formally.

The main difference between these two definitions is that in the first we must check every node with a property P that properly dominates α, while in the second it is enough to check only one node, just the first node with the property P that properly diominates α.

It can be easily seen that the command tests based on definition 1.2 are an important improvement in efficiency for computational applications.

Both versions (all) and (minimum) are used as command definitions in linguistic literature, so their equivalence also has linguistic consequences. For example, definition 1.3 of MAX-command from Barker/Pullum (1990) (which I formulate following Sells' definition of c-command, Sells (1987)) is equivalent to Definition 1.4, which has been proposed in Aoun/Sportich (1982).

1.3 DEFINITION. α MAX-commands β iff every maximal projection properly dominating α dominates β.

1.4 DEFINITION. α MAX-commands β iff the first maximal projection properly dominating α dominates β.

These definitions are special cases of definitions 1.1 and 1.2 for the property of being a set of maximal projections.

Before I formulate the general command definition in a formal way, I will now quote some other definitions from Barker/Pullum (1990).

1.5 DEFINITION. A relation R on a set N is **reflexive** iff aRa for all a in N; **irreflexive** iff $\neg$ aRa; **symmetric** iff aRb implies bRa; **asymmetric** iff aRb implies $\neg$ bRa; **antisymmetric** iff aRb and bRa implies a=b; **transitive** iff aRb and bRc implies aRc.
A relation R on a set N is called a **linear order** if it is reflexive, antisymmetric, transitive and has the following property (**comparability**): for every a and b in N, either aRb or bRa.

The following definition of a tree stems from Wall (1972).

1.6 DEFINITION. A **tree** is a 5-tuple
$$T=\langle N,L,\geq^D,<^P,LABEL\rangle, \text{ where}$$
N is a finite nonempty set, the nodes of T,
L is a finite set, the labels of T,
$\geq^D$ is a reflexive, antisymmetric relation on N, the **dominance relation** of T,
$<^P$ is an irreflexive, asymmetric, transitive relation on N, the **precedence relation** of T, and
LABEL is a total function from N into L, the labeling function of T, such that for all a, b, c and d from N and some unique r in N (**the root node of** T), the following hold:
(i)   The Single Root Condition: $r\geq^D a$
(ii) The Exclusivity Condition:
      $(a\geq^D b \vee b\geq^D a)\leftrightarrow\neg(a<^P b \vee b<^P a)$
(iii) The Nontangling Condition:
      $(a<^P b \wedge a\geq^D c \wedge b\geq^D d)\rightarrow c<^P d$

I will also use $>^D$ the proper dominance relation, which will be just like $\geq^D$ but with all pairs of the form $\langle a,a\rangle$ removed.

1.7 DEFINITION. If $\alpha M \beta$ we say that α is **the mother of** β, or α **immediately dominates** β, where $\alpha M\beta$ iff
$\alpha>^D\beta \wedge \neg\exists x [\alpha>^D x>^D\beta]$.

1.8 DEFINITION. A **property** P on a set of nodes N is a subset of N. If a node α satisfies P, I will write $\alpha\in P$ or $P(\alpha)$.

1.9 DEFINITION. The set of **upper bounds** for α with respect to a property P, written $UB(\alpha,P)$, is given by
$UB(\alpha,P)=\{\beta\in N: \beta>^D\alpha \wedge P(\beta)\}$.
Thus β **is an upper bound for** α if and only if it properly dominates α and satisfies P.

1.10 DEFINITION. Let X be any nonempty subset of a set of nodes N of a tree T. We will call an element α the smallest element of X and denote it as $minX$ if $\alpha\in X$ and for every node x $x\in X \rightarrow x\geq^D\alpha$. If X is an empty set, then $minX$ = the root node of T.
A set X is said to be **well-ordered** by $\geq^D$ if the relation $\geq^D$ is a linear order on X and every nonempty subset of X has a smallest element. For example, the set $\mathbb{Z}$ of integers with the usual ordering relation $\geq$ is well-ordered.

Now I can formally specify the meaning of the expression "the first node with a property P that properly dominates α" from the definition 1.2; it denotes the smallest element of the set $UB(\alpha,P)$, $minUB(\alpha,P)$. First I will show that this element always exists.
In set theory, it is a well-known fact that in any tree, a set of nodes that dominate a given node is well-ordered in the dominance relation (see Kuratowski/Mostowski (1976), for example). To be precise, for a given node α of a tree T, the set $UB(\alpha)=\{x\in T: x>^D\alpha\}$ is well-ordered. Hence, the set $UB(\alpha,P)=UB(\alpha) \cap P$ has a smallest element, which we denote $minUB(\alpha,P)$.
At this point we are ready to formally state command definitions 1.1 and 1.2.

1.11 DEFINITION (all). α **P-commands** β iff $\forall x (x\in UB(\alpha,P) \rightarrow x\geq^D\beta)$.

1.12 DEFINITION (minimum). α **P-commands** β iff $minUB(\alpha,P)\geq^D\beta$.

We say that P generates the P-command relation.

For example, we obtain the MAX-command relation (1.3) as a special case of Definition 1.11 if we take the set $\{\alpha \in N: LABEL(\alpha) \in MAX\}$ as a property P, where MAX is any set of maximal projections.

Definition 1.11 is the general command definition from Barker/Pullum (1990).

1.13 THEOREM. Definitions 1.11 (all) and 1.12 (minimum) are equivalent.

Proof. If a pair $<\alpha,\beta>$ fulfills the definition (all), then it also fulfills the definition (minimum), because $minUB(\alpha,P) \in UB(\alpha,P)$ if $UB(\alpha,P) \neq \emptyset$. If $UB(\alpha,P) = \emptyset$, then $minUB(\alpha,P) =$ the root node of T, so condition $minUB(\alpha,P) \geq^D \beta$ is also fulfilled.

Conversely, let a pair $<\alpha,\beta>$ fulfill the definition (min). This means that $minUB(\alpha,P) \geq^D \beta$.

We must show that $\forall x$ $(x \in UB(\alpha,P) \rightarrow x \geq^D \beta)$. If $UB(\alpha,P)$ is the empty set, then the claim is trivially fulfilled. If $UB(\alpha,P)$ is not empty, then let x be any element from $UB(\alpha,P)$. Then $x \geq^D minUB(\alpha,P)$. Since "$\geq^D$" is a linear relation on $UB(\alpha,P)$, it is transitive. Hence, $x \geq^D minUB(\alpha,P)$ and $minUB(\alpha,P) \geq^D \beta$ implies $x \geq^D \beta$.

2. AN INDEXING TECHNIQUE

I will now present an indexing mechanism which allows us to check any command relation in the sense of Definition 1.11 in a simple and straightforward way.

Let P be any property of nodes of a given syntax tree. The idea is the following: while the syntax tree is being built, there are special indices assigned to every node of this tree.

Generally, every node inherits indices from its mother.

Specifically, if $P(\alpha)$ holds for a node α, then a unique new index is put into the index set of α and the new index set of α obtained in this way will be inherited futher. This process is formally described in the following definition of functions indp and fp.

Letting T be any syntax tree, we define functions indp and fp from all nodes of T into finite subsets of N (the positive integers), whereby we can take finite subsets of any index set as a image of indp and fp.

2.1 DEFINITION. Let P be any property. The function $indp: N \rightarrow F(N) = \{\sigma \subseteq N: \sigma$ is a finite subset of $N\}$ is defined recursively as follows:

1^0 $indp(root(T)) = \{1\}$, where root(T) denotes the root node.

2^0 If α immediately dominates β, then
$indp(\beta) = indp(\alpha) \cup fp(\beta)$,

where **fp** is a function $fp: N \rightarrow F(N)$ which fulfills the following conditions:

1^1 If $\alpha \notin P$, then $fp(\alpha) = \emptyset$.

2^1 If $\alpha \in P$, then $fp(\alpha) = \{\tau\}$, for some unique index $\tau \in N$ $(\tau \notin \bigcup \{fp(\gamma): \gamma \in N$ and $\gamma \neq \alpha\})$.

The procedural aspect of this definition can be described as follows. First, the function fp assigns a set with a unique index to every node from P, and the empty set to every node which does not belong to P.

Then, for every node, the set it has been assigned by the function fp is added to the indices it inherits from its mother. The result is the value of the function indp.

Based on this description, it is easy to note the following facts.

2.2 FACT.
If $\gamma \geq^D \beta$, then $indp(\gamma) \subseteq indp(\beta) - fp(\beta)$.

2.3 FACT.
If $\gamma \in P$, then $\gamma \geq^D \beta$ iff $fp(\gamma) \subseteq indp(\beta)$.

Now I present the main theorem of this paper which gives a basis for efficient and simple implementations of P-command relations. Due to this theorem, we can check whether any P-command relation holds between two nodes by merely examining the subset relationship of corresponding index sets.

2.4 THEOREM. Node α P-commands node β iff $indp(\alpha) - fp(\alpha) \subseteq indp(\beta)$.

The proof, which makes use of equivalence Theorem 1.13, will be given at the end of this chapter.

To illustrate the P-command check based on the theorem above, I give an example for a MAX-command relation (1.3) which we obtain from general command definition 1.11 if we take the set $\{\alpha \in N: LABEL(\alpha) \in MAX\}$ as a property P, where MAX = $\{NP, VP, AP, PP, S\text{-bar}\}$ is a set of maximal projections (Sells 1987).

Let us consider the syntax analysis for sentence (2.5). In tree (2.6), the upper set of indices at every node corresponds to the value of the function fp for

this node and the lower set of indices corresponds to the value of the function indp for this node.

(2.5) **A friend of his saw every man with a telescope.**

We can see, for example, that the verb "saw" MAX-commands the prepositional phrase "with a

telescope", by verifying that indp("saw")-fp("saw")= {1,5} ⊆ {1,5,7}=indp("with a telescope"), or that "every man" does not MAX-command "his", by verifying that indp("every man")-fp("every man")= {1,5} is not a subset of indp("his")= {1,2,3,4}.

(2.6)
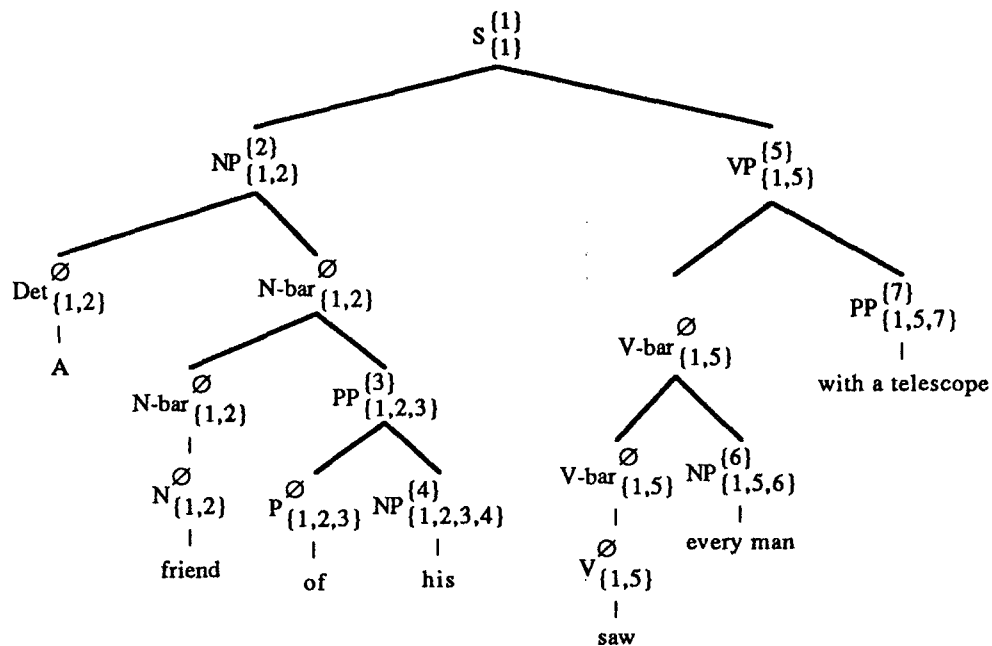
To do P-command tests in semantics, we merely need to extend functions fp and indp to semantic representations of every node. We can do this in the following way:

If α' is a semantic representation of a node α, then fp$(\alpha') = $ fp(α) and indp$(\alpha') = $ indp(α).

Now we can check, for two given semantic representations α', β', whether a P-command relation holds between the two corresponding nodes α, β, by examining the condition from Theorem 2.4 for α', β': indp(α') - fp$(\alpha') \subseteq$ indp(β'). (For more details see Latecki/Pinkal (1990).)

An important advantage of the indexing technique is that its applicability for checking command relations in semantics does not depend on an ispomorphism between syntactic and semantic structure, since the necessary syntactic information is encoded in indices. Therefore, this information can be moved to any required position in the semantic

structure together with the representation of a given node.

Definition 1.11 does not cover all cases of command relations which have been presented in linguistic literature, but there are only a few exceptions. One is the relation that is called c-command in Reinhart (1976; 1981, p.612; 1983, p.23):

2.7 DEFINITION. Node α c(onsitituent)-commands node β iff the branching node x_1 most immediately dominating α either dominates β or is immediately dominated by a node x_2 which dominates β, and x_2 is of the same category type as x_1. A node γ is a **branching node** iff there exists two different nodes x,y such that $\gamma Mx \wedge \gamma My$.

As T. Reinhart wrote, the intention of this definition is to capture c-command relations in cases S-bar over S or VP over VP. Hence, we can say (for

our purposes) that x_2 is of the same category type as x_1 if LABEL(x_2) = S-bar, LABEL(x_1) = S or LABEL(x_2) = LABEL(x_1) = VP.

This c-command definition allows the minimal upper bound to be replaced by another node, one node closer to the root, so this relation cannot be generated by any property, since this property must then depend on the node α. However, the condition of Definition 2.7 can be generated by a relation. In order to use a given relation R as generator, it is enough to replace the set of upper bounds UB(α,P) by the set UB(α,R)=$\{\beta \in N: \beta >^D \alpha \wedge \alpha R\beta)\}$, in general command definition 1.11. For detailed disscusion see Barker/Pullum (1990).

With an example of Reinharts c-commando relation, I want to show that **it is also possible to treat relational command definitions with the indexing technique.** Here I do not want to consider the treatment of the relational command definition with the indexing technique in the general case, because it would lead to a formal mathematical discussion without linguistic connections.

To specify a test for Reinharts c-command, we need merely to modify part 2^0 of the definition of the function indp in 2.1. The definition of the function fp together with the basis test condition given in Theorem 2.4 will be left unchanged. As a property P we take the set of branching nodes.

New part 2^0 of Definition 2.1 will be formulated in the following way:

$2^{0'}$ If α immediately dominates β and β is of the same category as α, then indp(β) = indp(α).

If α immediately dominates β and β is not of the same category as α then indp(β) = indp(α) $\cup$ fp(β).

The idea of this modification is that if a node β is of the same category as a node α, then β only inherits the indices from α. So, in this case, the new index from the set fp(β) does not influence the value of the function indp on β. I illustrate the indexing check for c-command definition 2.7 with the syntax analysis for the following example sentense from Reinhart (1983).

(2.8) **Lola found the book in the library.**

In tree (2.9), the upper set of indices at every node corresponds to the value of the function fp at this node and the lower set of indices corresponds to the value of the function indp at this node.

We can see, for example, that the subject of S, "Lola", c-commands the COMP in S-bar, by verifying that indp("Lola")-fp("Lola")=$\{1\}$ $\subseteq$ $\{1\}$= indp(COMP),
or that the object, "the book", c-commands the NP in PP, "the library", by verifying that
indp("the book")-fp("the book")=$\{1,4\}$ $\subseteq$ $\{1,4,7,8\}$= indp("the library").

(2.9)
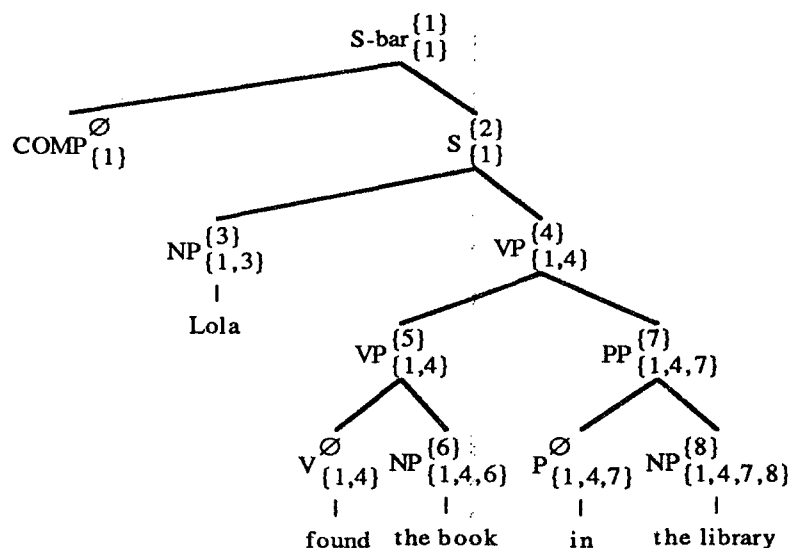
To conclude this chapter I give the proof of Theorem 2.4.

Proof of Theorem 2.4.

"$\Rightarrow$" Let α P-command β. If we denote γ=minUB(α,P) and indp(γ)=Σ, then indp(α) =$\Sigma \cup$ fp(α), because fp(x)=$\emptyset$, for every node x between α and γ.

Due to Definition 1.12 (minimum), the node γ also dominates β, hence $\Sigma \subseteq$ indp(β). So, we have the inclusion indp(α)-fp(α) $\subseteq$ indp(β).

"$\Leftarrow$" Now let indp(α)-fp(α) $\subseteq$ indp(β), for any two nodes α, β of some tree T, and let γ be any node from P that dominates α. indp(γ) $\subseteq$ indp(α)-fp(α), since γ dominates α (2.2).

From the transitivity of the inclusion relation, indp(γ) $\subseteq$ indp(β). This implies that fp(γ) $\subseteq$ indp(β). Due to Fact 2.3, the needed relation $\gamma \geq^D \beta$ holds, so α P-commands β.

3. CONCLUSIONS

I have presented an indexing technique which allows us to test all command relations that fulfill the definition of Barker and Pullum (1990). On an example of Reinharts c-command relation, I have also shown that it is possible to treat relational commands with this technique.

The indexing technique can be simply and efficiently implemented without any special requierments for the formalism used. Based on it, Millies (1990) has implemented tests for MAX-command, subjacency and government in a principle-based parser for GB (Chomsky 1981, 1982 and 1986).

It is also possible to use similar indexing processes to treat other linguistic phenomena in syntax as well as in semantics. Hence, the indexing technique has a wide spectrum of applications. For example, Latecki and Pinkal (1990) present an indexing mechanism which allows us to achieve the effects of "Nested Cooper Storage" (Keller 1988 and Cooper 1983).

ACKNOWLEDGMENTS

REFERENCES

Aoun, J. / Sportiche, D. (1982): On the Formal Theory of Government. *Linguistic Review* 2, 211-236.

Barker, Ch. / Pullum G. K. (1990): A Theory of Command Relations. *Linguistics and Philosophy* 13: 1-34.

Chomsky, N. (1981): Lectures on Government and Binding. Dordrecht: Foris.

Chomsky, N. (1982): Some Concepts and Consequences of the Theory of Government and Binding. MIT Press.

Chomsky, N. (1986): Barriers. *Linguistic Inquiry Monigraphs* 13, MIT Press: Cambrdge.

Cooper, R. (1983): Quantification and Semantic Theory. D. Reidel, Dordrecht.

Keller, W. R. (1988): Nested Cooper Storage: The Proper Treatment of Quantification in Ordinary Noun Phrases. In U. Reyle and C. Rohrer (eds.), *Natural Language Parsing and Linguistic Theories*, 432-447, D. Reidel, Dordrecht.

Kuratowski, K. / Mostowski, A. (1976): Set Theory. PWN: Warsaw-Amsterdam.

Latecki, L. / Pinkal, M. (1990): Syntactic and Semantic Conditions for Quantifier Scope. To appear in: Proceedings of the Workshop on "Processing of Plurals and Quantifiers" at GWAI-90, September 1990.

Millies, S. (1990): Ein modularer Ansatz für prinzipienbasiertes Parsing. *LILOG-Report* 139, IBM: Stuttgart, Germany.

Sells, P. (1987): Lectures on Contemporary Semantic Theories. *CSLI Lecture Notes.*

Reinhart, T. (1976): The Syntactic Domain of Anaphora. *Doctoral dissertation,* MIT, Cambridge, Massachusetts.

Reinhart, T. (1981): Definite NP Anaphora and C-Commands Domains. *Linguistic Inquiry* 12, 605-635.

Reinhart, T. (1983): Anaphora and Semantic Interpretation, University of Chicago Press, Chicago, Illinois.

Wall, R. (1972): Introduction to Mathematical Linguistics. Prentice Hall: Engelewood Cliffs, New Jersey.