# Binding pronouns with and without c-command

*Rogier Wuijts*

Bachelor Articial Intelligence major (7.5 ECTS)
Utrecht University
October 29, 2015

First Supervisor: Dr. Yoad Winter
Second Supervisor: Dr. Joost Zwarts

**Abstract**

This thesis aims to describe an alternative to c-command for the binding of pronouns and whether these alternatives are adequate and justified. I will present a theory on binding of pronouns with c-command by Büring to give context for comparison. The alternative to c-command consist of a variation on the binding conditions proposed by Bruening and a system with continuations to account for variable binding proposed by Barker. Both use a form of precedence instead of c-command and this is not only justified but also desirable when dealing with the binding of pronouns.

## Contents

# 1 Introduction

In 1976 Tanya Reinhart introduced a structural requirement on binding namely c-command, a relation between nodes in a parse tree. Since Reinhart defined it, c-command has become a fundamental concept in syntax, specifically in the binding of pronouns (Bruening, 2013, p. 1). Daniel Büring for example provided a proficient account of binding for anaphora, pronouns and referring expressions in his book Binding Theory (2005).

There are however also linguists who argue against c-command. Bruening (2013) thinks it is fundamentally flawed and shows a lot of empirical data where c-command doesn't predict if pronouns are being used correctly. He proposes a different relation, phase command, with an essential notion of precedence. He also argues that there is a difference between the grammatical rules of pronouns and the variable binding of pronouns that occur with quantified noun phrases and wh-phrases, something which was thought of as being the same thing. Bruening gives a theory for the former but the later is already done by another author, Chris Barker. Barker shows how variable binding without c-command can work using continuations. This leads me to the following question: How are these alternatives to c-command for the binding of pronouns justified and are these alternatives adequate ?

To answer this question, my thesis will be constructed as follows. In the second chapter I will present a system from Daniel Büring to give you an understanding what role c-command plays in these traditional theories. In the third chapter I will present arguments against c-command and an account of the binding conditions without c-command by Bruening. In this chapter I will also present the reasons why there is a difference between the binding conditions and variable binding. In the last chapter I will show how this variable binding without c-command can work, in a system of Barker that uses continuations.

Across this thesis I will use some basic semantics. All semantic denotations will be in lambda calculus. Also throughout this thesis I will also use an implicit function g from integers to entities when talking about sentences with co-refering nouns. Example: g(i) would refer to an entity. Also $\forall$i,j.(g(i)=g(j) $\Rightarrow$ i=j) holds. The index will be on the bottom right corner to indicate which person is referred to. Example: "Rogier$_1$ likes himself$_1$". In this case g(1) is Rogier.

# 2 Binding conditions and variable binding with c-command

## 2.1 Pronouns

An interesting thing about pronouns is that they do not directly refer. The reference from a pronoun must come from a context; Either the overall context in which for example I stretch out my arm and point to a certain guy or it comes from the context of the sentence or preceding sentences. An interesting question is how exactly a pronoun gets to co-refer with or bounded by a noun phrase. Besides the mechanisms involved which couple the binder with the bindee, there are a lot of grammatical restrictions on which type of pronoun you can use and which pronoun can be bound or co-refer. I will show this with a few examples:(*means the indicated interpretation is impossible or at the very least unwanted)

(1) Rogier$_1$ likes himself$_1$
(2) *Rogier$_1$ likes he$_1$
(3) Rogier$_1$ thinks he$_1$ is great
(4) *Rogier$_1$ thinks himself$_1$ is great
(5) He$_1$ thinks Rogier$_2$ is great
(6) *He$_1$ thinks Rogier$_1$ is great
(7) Everyone$_1$ likes his$_1$ mother
(8) *His$_1$ mother likes everyone$_1$

Most theories of binding distinguish three classes: anaphora[1] in sentence (1), pronouns in sentence (3), and R(eferring)-expressions in sentence (5). Each class has its own grammatical rules. In the Government and Binding Theory Chomsky formulated three binding conditions:

Condition A: An anaphor must be bound in its governing category.
Condition B: A pronoun must be free in its governing category.
Condition C: An R-expression must be free everywhere.
(Chomsky, 1981, p. 188)

The governor category in these conditions is the minimal domain containing it's governor and an accesible subject/SUBJECT(Haegeman, p. 223), which roughly translates to the clause. A binds B if they are both co-indexed and A c-command B where c-command is the structural relation introduced by Reinhart and defined as:
C-command:
-A does not dominate B,
-B does not dominate A, and
-The first (i.e. lowest) branching node that dominates A also dominates B
(Reinhart 1976, 8)

(9) [S [NP$_a$ Rogier$_1$] [VP [V likes] [NP$_b$ himself$_1$]]] sentence (1) with types and structure

In sentence (1) "Rogier$_1$" c-commands "himself$_1$" because NP$_a$ does not dominate NP$_b$ nor does NP$_b$ dominate NP$_a$ but the first node that dominates NP$_a$, namely S, also dominates NP$_b$.

Although these conditions and the Government and Binding Theory may be outdated, there are still a good reference point and are still used throughout the literature,

[1]anaphora in this context refers to reflexive pronouns, words like "himself" and "ourselves", and reproctrical pronouns, phrases like "each other" and "one another"

sometimes in a modified form. For example we could say sentence (2) is not well formed because condition B is violated. Sentence (4) is not well formed because condition A is violated. Sentences (6) is not well formed because condition C is violated.

There is something special about sentence (7) and (8): here the pronoun is bound by the quantified noun phrase (QNP). It's special because in sentence (1) and (3) the indexes co-refer; in other words they refer to the same entity. But that cannot be said of sentence (7). Ignoring condition C for just one moment we can think of sentece (1) as semantically equivalent to:"Rogier likes Rogier" but sentence (7) is not equivalent to "Everyone likes everyone's mother". In fact to index a QNP is a little weird in the first place. These types of noun phrases don't co-refer at all, they bind. The logic form of sentence (7) would be something like $\exists x.likes\ x\ (mother\ x)$. As stated by Reinhart binding and co-referring are the two semantic concepts that fall under the pre-theoretic concept of binding of pronouns. Sentence (3) can be a case of co-referring and sentence (7) is typical case of binding, but the distinction isn't that obvious since the bound version of (3) has the same semantic interpretation as the co-referring one which we will talk about in later sections. The reason sentence (8) is not well formed because it invokes a crossover violation and was one of the focus points of Chung-Chien Shan and Chris Bakker (2006) in which they explain their treatment to these kind of crossover violations.

Many theories of binding took the prinicples of the Government and Binding Theory regarding the three condition as the foundation of their theory and varied in what it means to be bound and what the governing category or domain precisely is. I will present one of these theories by Daniel Büring who was greatly influenced by Reinhart, to get you clear understanding of what such a binding theory looks like.

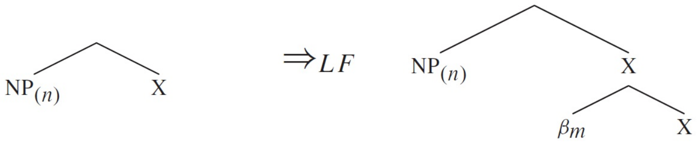
## 2.2 Bürings binding conditions using c-command

Büring uses c-command in his definition of semantic binding:

A binder prefix $\beta$ semantically-binds an NP if and only if
(a) $\beta$ and NP are coindexed
(b) $\beta$ c-commands NP
(c) there is no binder prefix $\beta$' which is c-commanded by $\beta$ and meets (a) and (b)

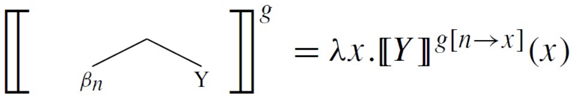
(Büring 2005, 86)

The variable $\beta$ here is a binder prefix that can be introduced by any noun phrase by applying the binder rule on it. When you have a noun phrase with a sister node X as in the picture, the binder rule replaces X with a node that has as left child: an introduced $\beta_{\mathrm{m}}$ and as right child: the original X. Since the noun phrase c-commands every node in X, any NP in X that is co-indexed is now semantically bound (assuming criteria c is not violated).



(Büring 2005, 109)

Sentence (1) for an example, Büring would transform this with the binder rule to: "Rogier$_1$ $\beta_1$[likes himself$_1$]". And the subsequence, "$\beta_1$[likes himself$_1$]", can be interpreted with the Binder Index Evaluation Rule (BIER):



(Büring 2005, 85)

Function g is a function from indexes to entities and the result of BIER is that all noun phrases with index n in Y are now replaced by variable x. With the lambda abstraction this variable gets bound at the logical form. The subsequence, "$\beta_1$[likes himself$_1$]", applied to BIER will result in $\lambda x.$*(likes x) x*. A function that characterizes the set of all people who like themselves (ignoring the fact himself only applies to men). Combined with "Rogier" we get the desired semantic result *(likes Rogier) Rogier*.

According to Büring not only QNP's but all full NP's should not even have an index, QNP's don't refer and referring expressions already refer. "Rogier" already refers to *Rogier* and it would be weird to say Rogier$_1$ where g(1) $\neq$ *Rogier*. So he leaves out indexes on R-expressions altogether and without an index they cannot be bound. Because of this line of reasoning Büring will drop binding condition C since it bears no meaning in his theory any longer. Instead, he introduces another principle to account for what otherwise would have been a condition C violation. Büring introduces the Have Local Binding rule(HLBR) that forces you to bind certain elements. This rule is introduced because a sentence with a bound pronoun and a sentence with a co-refering pronoun semantically can mean the same thing.

(10) Rogier thinks he$_1$ is great (Where g(1) = Rogier, he is co-refering with Rogier)
(11) Rogier $\beta_1$ [thinks he$_1$ is great] (he is bounded by Rogier)

In order to have rules that apply to both sentences Büring forces you to rewrite one to the other. Inspired by Reinhart's co-reference rule which states that: "$\alpha$ cannot corefer with $\beta$ if an indistinguishable interpretation can be generated by replacing $\alpha$ with a variable bound by $\beta$ " (Büring 2005, 166), Büring introduces :

Have Local Binding:
"For any two NPs $\alpha$ and $\beta$, if $\alpha$ could bind $\beta$ $\alpha$ must bind $\beta$, unless that changes the interpretation."
(Büring 2005, 129)

Since in sentence (10) "he$_1$" can be bound by "Rogier" and since the bound version doesn't differ in interpretation it must be bound as in sentence (11). According to Büring this rule also implies that sentences that would normally violate condition C like "$\text{He}_1$ thinks $\text{Rogier}_1$ is great" are ruled out because the same interpretation can be achieved by "$\text{He}_1$ $\beta$i [thinks $\text{he}_1$ is great]" and thus BIER replaces condition C. Büring does copy condition A and B with slight alterations:

(A) A reflexive pronoun must be semantically bound in its domain.
(B) A non-reflexive pronoun must be semantically free in its domain.
(Büring 2005, 129)

The theory of Büring presented so far captures the basics of the theory of binding with respect to refexive pronouns, non-reflexive pronouns and referring expressions. But there is still an interesting topic that hasn't been covered yet:crossover violations.

## 2.3 Crossover

The term "crossover" refers to a wh-noun phrase that crosses over a pronoun and as a result cannot bind the pronoun. For example two wh-sentences that had movement and left a wh-trace behind.

(12) $\text{Who}_1$ did $\text{he}_1$ think $t_1$ is tall?
(13) $\text{Who}_1$ did James thinks $t_1$ loves $\text{his}_1$ mom?

In sentence (12) the wh-trace crosses over the pronoun, making the co-indexed reading impossible. But wh-movement where the wh-trace binds the pronoun can work just as fine as in sentence (13). The way these traces must be interpreted is almost the same as pronouns and thus also the same as with the binder rule and BIER. The wh $\alpha$ will leave a trace t behind and places a trace bind index $\mu_{\text{i}}$ at $\alpha$'s sister node. The interpretation rule is similar to BIER and is called The Movement interpretation rule. In the sister node of binder $\mu_{\text{i}}$ all $t_{\text{i}}$ will be replaced with a variable that is abstracted over. The normal binding rules don't rule out these violations of (12) so some additions need to be made. Büring introduces one solution where the binder rule can only be applied if the binder don't have any movement. This will rule out the incorrect evaluation of (12)."Who $\beta_1$ $\mu_1$ did $\text{he}_1$ think $t_1$ is tall ?". "Who" is trying to bind $\beta_1$ but "Who" has moved so this is incorrect. Sentence (13) can be interpreted just fine since "$\text{his}_1$" is bound by $t_1$ and not by "Who" in "Who $\mu_1$ $t_1$ $\beta 1$ calles $\text{his}_1$ mom".

Not only wh-movement can trigger crossover violations. Although a quantifier can raise to the front of a expression in a logical form and take scope over it, it cannot bind the pronouns. Büring handles this the same way as wh-movement. When the QNP raises, it leaves a trace behind that can be abstracted over, but $\beta_1$ cannot bind "Every desk" since it has moved.

(14) *A picture of its owner was standing on every desk.
LF(14):*Every desk $\beta_1[\mu_1$[a picture of $its_1$ owner was standing on $t_1$]]
(Büring 2005, 166)

# 3 Binding conditions without c-command

## 3.1 Against c-command

As you can see in the previous chapter in almost all the work by Büring there is a notion of c-command. Not only Büring but many linguists use c-command. But there are also theories that favor another relation and take precedence into account (Bruening 2013, p. 2). Reinhart had always rejected precedence. She showed empirical data concerning fronted phrases that ruled out precedence as an indicator (sentence (17) and (18)). In all these cases c-command made the right prediction. As we will see next, Bruening tries to refute Reinharts arguments against precedence and in favor of c-command. Bruening (2013) pleas to drop c-command because he argues that it is fundamentally flawed. There are numerous conflicts between tests for constituency and c(onstituent)-command. Specifically c-command as a requirement for Binding Theory has a lot of problems concerning prepositions, VP Adjuncts, IP Adjuncts and coordination according to Bruening.

(15)* Sue spoke to $him_1$ about $Bill_1$'s mother.
(16)* Mary played quartets with $them_1$ at [John and $Sue]_1$'s party.
(Bruening 2013, 6)

In the sentences above the pronoun does not c-command the NP, therefore condition C violation does not occur if you base it on c-command. To account for these cases some authors have been proposing modfications to c-command but Bruening says that these modifications can account for some but not all the problems. Reinharts argument against precedence was that precedence could not explain why fronted phrases were grammatical or ungrammatical.

(17) Near $him_1$, $Dan_1$ saw a snake.
(18) * Near $Dan_1$, $he_1$ saw a snake.
(19) * Near $Dan_1$, I heard that $he_1$ saw a snake.
(20) Near $Dan_1$, I saw $his_1$ snake.
(Bruening 2013, 19-20)

Reinhart argues that in these two examples c-command is the relevant notion, not precedence. But as Bruening explains you can modify these sentences in such a way that the pronoun doesn't c-command the noun phrase as in sentence (19) but it results in an ungrammatical sentence. He argues that reconstruction is the anwser here. Reinhart dissmissed the idea of reconstruction the PP because reconstructing it in its surface position or its base position would not explain the object-subject asymmetry, sentence

(18) is wrong but sentence (20) is well formed. Bruening argues that these fronted PPs have to be interpreted with the verb phrase, essentially reconstructing them behind the noun phrase. Sentence (18) and (19) are not well formed because they commands[2] and proceed "$\text{Dan}_1$", which will trigger a condition C violation. This also solves the problem of subject-object asymmetry since the PP is reconstructed behind the subject but before the object. He concludes there is no valid argument against precedence and that precedence is even required in most cases.

(21) Penelope cursed $\text{Peter}_1$ and slandered $\text{him}_1$.
(22) * Penelope cursed $\text{him}_1$ and slandered $\text{Peter}_1$.
(Bruening 2013, 12)

In sentence (21) and (22) both the subject and the object c-command each other. In both cases they will violate condition C with c-command but intuitive we would say sentence (21) is correct and (22) is wrong. The only way to get rid of this symmetrical relation is with a notion of precede.

### 3.2 Bruenings Binding conditions with Precede and Phase command

Bruening introduces a new command, namely phase-command. This together with precedence is his precede-and-command what will be the leading relation for the binding conditions of Bruening. He argues that c-command takes all nodes into account, which result in problems with, amongst other things, prepositions. He proposes to only use certain nodes and he calls these nodes phases. Phases are maximal clauses (CP), maximal verb phrases (vP) or maximal nominal projections (NP). These particular nodes are of special importance to syntax "being involved in cyclicity, spellout, successive-cyclic movement, and locality constraints on agreement. " (Bruening 2013, 2)

> Phase-command: X phase-commands Y iff there is no ZP, ZP a phasal node, such that ZP dominates X but does not dominate Y.
> (Bruening 2013, 2)

Bruening claims that phase-command is a leading principle in syntax. He derives his version of the binding conditions based on the view that grammar in language is constructed from left to right. Condition A, B and C are not a form of variable binding but of co-reference. He argues that the principles involved in the binding conditions and variable binding are two different things. The conditions that give rise to cross over violations with QNP's and wh-movement differ from those that give rise to binding condition violations.

Bruening creates his take on condition C by using two discourse sets: set D that consists of all referents in the current discourse, a set C consisting of all referents currently being processed and a set A constisting of all local referents currently being processed.

[2]Bruenings phase-command not c-command

Conditions A and B will be defined with set A and condition C will be defined with set C as I will describe later. There is also a transitioning principle.

> Processing Principle:
> Move discourse referent R denoted by NP N out of active set C or local set A and into set D at the right edge of a phasal node that dominates N.
> (Bruening 2013, 33)

This principle captures the notion of precede and command. With this principle he defines a condition C that is based on Minimize Restrictors of Schlenker(2004).

> Condition C :
> A definite description of the form **the A** may not refer to a discourse referent in active set C if **A** could be dropped without affecting either (i) the denotation of the description or (ii) its various pragmatic effects.
> (Bruening 2013, 33)

It should be noted that a proper name also falls under the definition of definite descriptions according Bruening since all proper names have an implicit definite article. Consider the sentence repeated from the previous chapter:

(6) *He$_1$ thinks Rogier$_1$ is great
(23) Her$_1$ assistant doesn't like the teacher$_1$'s students.

As the sentence is processed from left to right first "He$_1$" is put into set C then "Rogier$_1$" is encountered and it is of the form **the A**. It may not co-refer with "He$_1$" in set C because **A** could be dropped, namely "He$_1$ thinks he$_1$ is great". Sentences (23) can be processed in the same way. A sentence is ungrammatical due to a condition C violation if the R-expressions refers to some NP in set C and the **the** can be dropped without effecting the denotation or it's pragmatic effects. In sentence (23) "the teacher" does not refer to a discourse referent in C because "Her$_1$" is in set D. " Her$_1$" was moved from C to D when the phasal node "Her$_1$ assistant" was being processed and thus sentence (23) is grammatically correct.

The Minimize Restrictors of Schlenker(2004) and the Have Local Binding rule of Büring are somewhat pragmatic. If it can be said otherwise without R-expression and without losing the interpretation you should say it with pronouns. Schlenker, and therefore Bruening with his own condition C, also claim that usage of a R-expression can be justified if the R-expression brings disambiguation. Condition C therefore cannot only be structural.

For condition A and B Bruening introduces another discourse set A, which will be even more local than active set C. When elements are encountered they are placed in set A and when the local domain of another verb starts these elements would be moved from A to C. The condition will also move elements from A to D when it is at the right edge of a phasal node. Now Bruening defines condition A and B as follows:

Binding Condition A:
If a newly processed NP N has the form of a local anaphor, it must denote a discourse referent in set A.
Binding Condition B:
If a newly processed NP N is to be interpreted as denoting a discourse referent R already in set A, then N must have the form of a local anaphor.
(Bruening 2013, 38)

The reformulated condition A is very similar to the original condition but condition B is not. You could interpret (to highlight similarities) *a pronoun must be free in its domain* as *if it is not free in its domain it cannot be pronoun and it must be a anaphor.* *"Rogier$_1$ likes him$_1$", would be ungrammatical because when processed from left to right, "Rogier$_1$" is in the A set when "he$_1$" is processed. That will trigger a condition B violation because "he$_1$" does not have the form of a local anaphor. "Rogier$_1$ thinks himself is great$_1$" would trigger a condition A violation since "himself" has the form of a local anaphor but Rogier is not in the set A. Rogier is not in set A because when "is" is processed Rogier is moved from A to B since "is" is of another predicate than thinks. Bruening version gives an account of the binding conditions without c-command but with his own notion of phase-command and precedence.

### 3.3 Difference in the Binding conditions and variable binding

Bruening also makes another claim namely that condition C does not relate to quantifier scope or variable binding because the two principles obey different laws.

(24a) Rosa is kissing him$_1$ passionately in Ben$_1$'s high school picture.
(24b) Rosa is kissing every boy$_1$ passionately in his$_1$ high school picture.
(25a) People worship him$_1$ in Kissinger$_1$'s native country.
(25b) People worship every UN Secretary-General$_1$ in his$_1$ native country.
(26a) So many people wrote to him$_1$ that Brando$_1$ couldn't answer them all.
(26b) So many people wrote to every actress$_1$ that she$_1$ couldn't answer them all.
(Bruening 2013, 35-36)

Bruening claims if condition C and quantified binding are related, replacing the QNP with a normal NP, should result in binding condition violations. But as you can see for yourself, it does not. Not only QNP but also variable binding from wh-phrases deviate from the binding conditions and also require no c-command. The claim that variable binding and co-reference do relate to each other has been made by Reinhart and much of her predecessors including Büring. Bruening claims the contrary and this is precisly what Chung-Chien Shan and Chris Barker did. They showed that they could present a system that can bind variables and account for various phenomena like crossover violations without the use of c-command but with a notion of precedence, in the way of a evaluation order. These sentences below from Barker (2012) show that in cases when the QNP is embedded inside various clauses, binding occurs without a relation of c-command:

(27) [Everyone$_i$ 's mother] thinks he$_i$ 's a genius. (possessive DP)
(28) [Someone from every$_i$ city] hates it$_i$ . (nominal complement)
(29) John gave [to each$_i$ participant] a framed picture of her$_i$ mother. (prepositional phrase)
(30) We [will sell no$_i$ wine] before it$_i$ s time. (verb phrase)
(31) [After unthreading each$_i$ screw], but before removing it$_i$ . (temporal adjunct)
(32) The grade [that each$_i$ student receives] is recorded in his$_i$ file.(relative clause)
(Barker 2012, 28)

# 4 Variable binding without c-command

In the next chapter I will introduce a binding mechanism that can bind pronouns. Before I can continue with that I must first introduce the system in which these binding mechanism take place, namely a system that uses continuation.

## 4.1 Continuation

In Computer Science continuation is the remaining part of the computation at any time in the computation. By representing this computation in an abstract representation a programming language has access to its own computation process. For instance first-class-continuation is used to capture this computation and return, at a later time, to the execution state of that time. This capturing of the execution state can be used to deal with exceptions. Also continuation gives an order independent way to reason about evaluation order in a formal language. This is one of the reasons continuation, delimited to be precise, is used by Barker(2012). The basic principle is his continuation hypotheses:

> The continuation hypothesis: some natural language expressions denote functions on their continuations, i.e., functions that take their own semantic context as an argument.
> (Barker 2012, 1)

What Barker means by this is easily illustrated:

(33) John saw everyone in the shower
(34) John saw bob in the shower
(35) *in-the-shower((see b) j)*

(36) $\forall e.$ *in-the-shower((see e) j)*
(37) $\lambda e.$ *in-the-shower((see e) j)*
(38) $\lambda K. \forall e. K\ e$

We have two sentences (33) and (34) where sentence (33) has a full noun phrase, "bob", and sentence (34) has a quantified noun phrase, "everyone". Their semantic representation is (35) and (36). But here they differ even more, not only regarding the object but in sentence (36) everyone quantifies over the whole lambda term. *Everyone* takes scope over its continuation. With other words everyone here denotes a function on its continuation. This function is shown in (38) where variable K stands for continuation. In both sentences (33) and (34) bob and everyone have the same continuation, they have the same context. "John saw [ ] in the shower "or as in (37) in lambda terms. Barker uses combinatory categorial grammar with a small number of type-shifters for its continuation based grammar. In CCG a function can combined with its argument left as in (39) or right as in (40).

(39) $B/A::(\alpha \rightarrow \beta) \otimes A::\alpha \Rightarrow B::\beta$
(40) $A::\alpha \otimes A\backslash B::(\alpha \rightarrow \beta) \Rightarrow B::\beta$

However linear adjacency either left or right for everyone in sentence (33) does not suffice as everyone is in its own argument. Therefore Barker introduces another syntactic notion, surrounding (41) or surrounded by (42).

(41) $A \backslash\!\!\backslash B$ (becomes an B when it surrounds a A) sem type $\alpha \rightarrow \beta$ just as $A\backslash B$
(42) $B /\!\!/ A$ (becomes an B when it is surrounded by A) sem type $\alpha \rightarrow \beta$ just as $B/A$

In our case "John saw [ ] in the shower " would become $NP \backslash\!\!\backslash S$ with bob being the missing NP but "everyone" would become $S_1 /\!\!/ (NP \backslash\!\!\backslash S_2)$, it becomes a $S_1$ when it is surrounded by that what becomes $S_2$ when is surrounds a NP . This can be interpreted as a word that functions syntactically or locally as an NP takes scope over $S_2$ and creates $S_1$. The semantic type is $(e \rightarrow t) \rightarrow t$ just as a generalized quantifier would have been with a syntactic type of $S/(NP\backslash S)$.

We can create this type by using the type shift operators:

Lift(U)[3] : $B /\!\!/ (A \backslash\!\!\backslash B)/A\ \lambda x.\lambda f.f\ x$
Lower(D): $A/(A /\!\!/ (S^4 \backslash\!\!\backslash S))\ \lambda f.f\ (\lambda x.x)$

And a combination rule for elements of the form $X /\!\!/ ((A/B) \backslash\!\!\backslash Y)$ and $Y /\!\!/ ((B \backslash\!\!\backslash Z)$ to make $X /\!\!/ ((A) \backslash\!\!\backslash Z)$

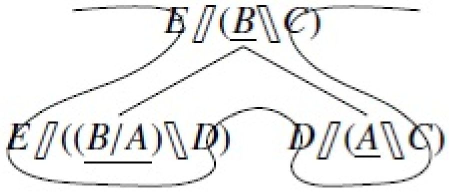
[3] sometimes referred to as "UP" in "Explaining crossover and superiority as left-to-right evaluation" but I will use Lift as in "Continuation and Natural Language" . When using another lift like lift: $A \rightarrow B/(A\backslash B)$ where it is obviously based on, I will make it explicit.

[4] S in this case is not a meta-variable but the S as in Sentence

Scope rule = $(X /\!\!/ ((A)\backslash\!\!\backslash Z))/(Y /\!\!/ ((B\backslash\!\!\backslash Z))/(X /\!\!/ ((A/B)\backslash\!\!\backslash Y))$
$\lambda L.\lambda R.\lambda K.\ L(\lambda l.R(\lambda r.K(\ l\ r)))$

For example: Let element M be such that it functions locally as type A takes scope over type Y and returns type X and let element N be such that it functions locally as type A\B takes scope over type Z and returns type Y. Rule S applied to element M and N would result in an element that functions locally as a B takes scope over Z and returns a X.

It is important to note that in order for the scope rule to be applied, the arguments must have a specific form, as this picture from Barker(2006) shows.




(Barker 2006, 9)

To show how this rule works I give you a simple example. Suppose we have "Rogier"::NP and "sleeps"::NP\S. In normal CCG we can just combine the two and get semantic meaning of *sleeps r* where *sleeps* is a predicate and *r* an entity. But for the sake of explaining the system we will lift the whole system to a higher continuation level. First we will use the normal lift on Rogier[5]. We then apply the continuous lift with S to both the lifted Rogier and sleeps. We get Rogier:: $S /\!\!/ ((S/(NP\backslash S))\backslash\!\!\backslash S)$and sleeps :: $S /\!\!/ ((NP\backslash S)\backslash\!\!\backslash S)$ when we combine these two with the scope rule we get $S /\!\!/ (S\backslash\!\!\backslash S)$ which is of such a form that we can use the type shifter Down and result in a S. Semantically it will result in a very long lambda term which I will beta reduce now:

D (S(U (L Rogier)) (U sleeps))[6]
$\lambda F.F\ \lambda x.x\ (\lambda L.\lambda R.\ \lambda k.\ L\ \lambda l.R\ \lambda r.k(lr)\ (\lambda x.\lambda F.F\ x\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}))\ (\lambda x.\lambda F.F\ x\ \text{sleeps})) \rightsquigarrow \beta$
$\lambda L.\lambda R.\ \lambda k.\ L\ \lambda l.R\ \lambda r.k(lr)\ (\lambda x.\lambda F.F\ x(\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}))\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda x.x \rightsquigarrow \beta$
$\lambda R.\ \lambda k.(\lambda x.\lambda F.F\ x\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}))\ \lambda l.R\ \lambda r.k(lr)\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda x.x \rightsquigarrow \beta$
$\lambda k.(\lambda x.\lambda F.F\ x\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}))\ \lambda l.\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda r.k(lr)\ \lambda x.x \rightsquigarrow \beta$
$(\lambda x.\lambda F.F\ x\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}))\ \lambda l.\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda r.\ \lambda x.x\ (lr) \rightsquigarrow \beta$
$\lambda F.F\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier})\ \lambda l.\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda r.\ \lambda x.x\ (lr) \rightsquigarrow \beta$
$\lambda l.\ (\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda r.\ \lambda x.x\ (lr)\ (\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier}) \rightsquigarrow \beta$
$(\lambda x.\lambda F.F\ x\ \text{sleeps})\ \lambda r.\ \lambda x.x\ ((\ \lambda x.\ \lambda F.\ F\ x\ \text{Rogier})\ r) \rightsquigarrow \beta$

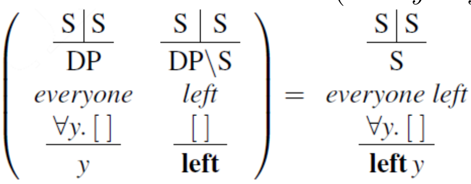
[5]Due to the fact this scope rule presented by the paper of Barker only uses A/B and not B\A, this cumbersome method will not be used later when I use tower notation
[6]We could also replace Rogier in the logical form with a everyone we would get $\forall x.\text{sleeps } x$.

λF.F sleeps λr. λx.x (( λx. λF. F x Rogier)r r )⇝ $\beta$
λr. λx.x (( λx. λF. F x Rogier) r) sleeps ⇝ $\beta$
λx.x (( λx. λF. F x Rogier) sleeps) ⇝ $\beta$
( λx. λF. F x Rogier) sleeps ⇝ $\beta$
(λF. F Rogier) sleeps ⇝ $\beta$
sleeps Rogier

We can organize a little party because we derived the meaning to be *sleeps Rogier* in one of the most cumbersome ways ever. Luckily for us Barker introduced a simpler notation for this, although this notation lets you easily forget that the scope rule is significant in his theory. The scope rule has a bias towards left right evaluation; it forces to first evaluate the left sides before evaluating the right side. Although it may seem cumbersome, it is in fact strict procedure that grants precise control over order evaluation.

## 4.2 Tower notation

The simpler notation is the tower notation introduced in Barker(2012). It represents the syntactic form (in the case of everyone S $/\!\!/$(DP$\backslash\!\!\backslash$S) and the semantical lambda representation with continuations ($\lambda K.\forall y.K\ y$ in the case of everyone).




(Barker 2012, 14)

So the left corner corresponds to what the expression returns and the right corner corresponds with the type it takes scope over and the bottom one corresponds with how it acts locally. Not only does the syntax have a more readable notation the semantic representation gets an improvement as well. This semantic tower notation corresponds with $\lambda K.\forall x.K\ x$ for everyone and $\lambda K.K\ \mathit{left}$ for left. But this notation really excels in removing the complexity of the scope rule as you can now simply combine the two if they have the right syntactic form, just like the picture of the scope rule. At the part above the line, you can fill the right section into to gap of the left section (order sensitivity). The part below the line corresponds to what the local type of syntax is. In this case DP\S is the function and DP argument so it combines to *left y*.

This system can handle scope taking but it does not leave room for scope ambiguity yet. Sentences like "Someone loves everyone" will only give the absolute reading[7] in the part of the system we introduced. Because the scope rule has a bias

[7]Absolute reading is that there is exisit someone that loves everybody($\exists x \forall y\ \mathit{loves}\ x\ y$). The relative reading is that everybody is loved by someone($\forall y \exists x\ \mathit{loves}\ x\ y$)

towards left right evaluation, it would not be right to change the bias towards right left evaluation to get the relative reading cause that would cause ungrammatical sentences to be derived. Instead Barker lifts the whole structure to another continuous level. If NP is continuous level 0 (pure) then S ⫽(NP⧹S) is called level 1 and S ⫽(S ⫽(NP⧹S)⧹S) level 2. Barker uses U and (S (UU)) to create different continuation lifts. A quantifier that I lifted with S (U U) takes scope over the one using U. When combined with everyone:: S ⫽(NP⧹S) we get S (U U) S ⫽(NP⧹S) = (S ⫽((B ⫽(NP⧹B))⧹S)) what is different from U S ⫽(NP⧹S) = (B ⫽((S ⫽(NP⧹S))⧹B)). Or in tower notation:

$$\begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{\mathrm{DP}} \\ \textit{everyone} \\ \frac{\forall y.[\,]}{y} \end{array} \quad \overset{\text{LIFT}}{\Rightarrow} \quad \begin{array}{c} \frac{B\,|\,B}{\mathrm{S}\,|\,\mathrm{S}} \\ \hline \mathrm{DP} \\ \textit{everyone} \\ \frac{[\,]}{\forall y.[\,]} \\ \hline y \end{array} \qquad \begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{\mathrm{DP}} \\ \textit{everyone} \\ \frac{\forall x.[\,]}{x} \end{array} \quad \overset{\text{LIFT}}{\Rightarrow} \quad \begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{B\,|\,B} \\ \hline \mathrm{DP} \\ \textit{everyone} \\ \frac{\forall x.[\,]}{[\,]} \\ \hline x \end{array}$$

(Barker 2012, 42-43)

The one that takes scope at a higher continuous level takes scope over a lower continuous level and with these notions we make a difference between the relative reading and the reverse reading.

$$\begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{\mathrm{S}\,|\,\mathrm{S}} \\ \hline \mathrm{DP} \\ \text{someone} \\ \frac{[\,]}{\exists x.[\,]} \\ \hline x \end{array} \left( \begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{\mathrm{S}\,|\,\mathrm{S}} \\ \hline (\mathrm{DP}\backslash \mathrm{S})/\mathrm{DP} \\ \text{loves} \\ \frac{[\,]}{[\,]} \\ \hline \textbf{loves} \end{array} \quad \begin{array}{c} \frac{\mathrm{S}\,|\,\mathrm{S}}{\mathrm{S}\,|\,\mathrm{S}} \\ \hline \mathrm{DP} \\ \text{everyone} \\ \frac{\forall y.[\,]}{[\,]} \\ \hline y \end{array} \right)$$

(Barker 2012, 44)

Since everyone operates at higher Continuous level in this tower notation, we get the relative reading: $\forall y \exists x$ *loves x y*.

## 4.3 Binding of pronouns in Barkers system

Now we have some notion of the system Barker uses and we can introduce that what really matters to this paper, pronouns. For pronouns Barker introduces another connective namely ▷. A▷B is basically a B except that it has an unbounded pronoun of category A in it. If a Sentence S would have an unbound pronoun NP in it, the type should be NP▷S. The pronoun can either be bound by a noun phrase, normal or quantified, or it could come from the pragmatic context. In that latter case not much can be said of it expect of course that it is of type NP▷S. It wouldn't be a theory of binding if the unbounded pronoun couldn't be bound by something so Barker introduces an operator

Bind:

$\text{Bind} = (\text{B} /\!/ (\text{NP} \backslash\!\!\backslash (\text{NP} \triangleright \text{A}))) / (\text{B} /\!/ (\text{NP} \backslash\!\!\backslash \text{A}))$
$\lambda\text{X}.\lambda\text{K}.\text{X}\ (\lambda\text{y}\ (\text{K y})\ \text{y})$
(Barker 2006, 13)

In this definition of bind the y variable gets doubled so the continuation of the word that binds the pronoun feeds the semantical interpretation of its continuation two times, just like the Binder Index Evaluation rule of Daniel Buring where the NP is not only an argument for function but also replaces any bind variables. In the case that everyone binds a pronoun the type would be:

$\text{Bind S} /\!/ (\text{NP} \backslash\!\!\backslash \text{S}) = (\text{B} /\!/ (\text{NP} \backslash\!\!\backslash (\text{NP} \triangleright \text{A})))$
$\lambda\text{X K}.\text{X}\ (\lambda\text{y}\ (\text{K y})\ \text{y})\ (\lambda\text{K'}.\forall\text{x}.\text{K' x}) \rightsquigarrow \beta\ \lambda\text{K}.\ (\forall\text{x}.(\text{K x})\ \text{x})$

Or in tower notation:

$$\frac{\text{S} \mid \text{S}}{\text{DP}} \quad \textsc{bind} \quad \frac{\text{S} \mid \text{DP} \triangleright \text{S}}{\text{DP}}$$
$$\textit{everyone} \quad \Rightarrow \quad \textit{everyone}$$
$$\frac{\forall x.[\,]}{x} \qquad\qquad \frac{\forall x.[\,]x}{x}$$

(Barker 2012, 26)

Now we got all things in place to analyze a sentence with a bounded pronoun using Barker's system. Take for an example "Everyone$_1$ loves his$_1$ mother". Here "Everyone$_1$" is binding "his$_1$". *Everyone* applied to the type shifter bind results in type $\text{S} /\!/ (\text{DP} \backslash\!\!\backslash (\text{DP} \triangleright \text{S}))$ and "his$_1$" is a pronoun so it gets the type $\text{DP} \triangleright \text{S} /\!/ (\text{DP} \backslash\!\!\backslash \text{S})$. Mother would just be a DP modifier and thus gets the type $\text{DP} \backslash \text{DP}$ but in order to combine with his it gets lifted with type S resulting in $\text{S} /\!/ ((\text{DP} \backslash \text{DP}) \backslash\!\!\backslash \text{S})$. Loves is of type $(\text{DP} \backslash \text{S})/\text{DP}$ and cannot be lifted with S because then the sequence wouldn't be of the correct form in order for the scope rule to work. Loves is therefore lifted with $\text{DP} \triangleright \text{S}$ resulting in $\text{DP} \triangleright \text{S} /\!/ ((\text{DP} \backslash \text{S})/\text{DP}\ \backslash\!\!\backslash \text{DP} \triangleright \text{S})$. We can now combine this using the tower notation:

$$\begin{array}{c} \dfrac{S \mid DP\triangleright S}{DP} \\ \textit{everyone} \\ \dfrac{\forall x.[\,]x}{x} \end{array} \left( \begin{array}{c} \dfrac{DP\triangleright S \mid DP\triangleright S}{(DP\backslash S)/DP} \\ \textit{loves} \\ \dfrac{[\,]}{\mathbf{loves}} \end{array} \left( \begin{array}{c} \dfrac{DP\triangleright S \mid S}{DP} \\ \textit{his} \\ \dfrac{\lambda y.[\,]}{y} \end{array} \quad \begin{array}{c} \dfrac{S \mid S}{DP\backslash DP} \\ \textit{mother} \\ \dfrac{[\,]}{\mathbf{mom}} \end{array} \right)\right)$$

$$= \begin{array}{c} \dfrac{S \mid S}{S} \\ \textit{Everyone loves his mother} \\ \dfrac{\forall x.(\lambda y.[\,])x}{\mathbf{loves}\,(\mathbf{mom}\,y)\,x} \end{array} \quad \overset{\text{LOWER}}{\Rightarrow} \quad \begin{array}{c} S \\ \textit{Everyone loves his mother} \\ \forall x.(\lambda y.\mathbf{loves}\,(\mathbf{mom}\,y)\,x)x \end{array}$$

(Barker 2012, 27)

Because the elements have the correct syntactic form in relation to each other they can be combined, but that is not enough. We end up with something not of the form S but of the form $S /\!\!/ (S \backslash\!\!\backslash S)$. It must therefore be lowered. The Lower rule can only be applied to element of the form $(\_ /\!\!/ (S \backslash\!\!\backslash S))$. In this case it is no problem and we get the desired derivation. A thing to note is that in order for a QNP to bind the pronoun there must be a chain of DP▷S at the continuation level where the QNP is trying to bind the pronoun (in this case level one). As you will see next, most of the time we want to derive an ungrammatical sentence the derivations stops at the point where the syntactic type is of such form that it can't be lowered.

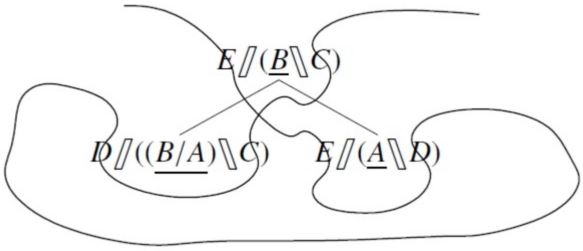
Crossover violations arise when the pronoun that is going to be bound by a QNP, precedes this QNP in evaluation order. Sentences like "His$_1$ mother loves everyone$_1$." have a weak crossover violation and are impossible to derive in Barkers current system. When we give the elements the same treatment as we did to the later sentence, we would get something like this:

$$\left( \begin{array}{c} \dfrac{DP\triangleright S \mid S}{DP} \\ \text{his} \end{array} \quad \begin{array}{c} \dfrac{S \mid S}{DP\backslash DP} \\ \text{mother} \end{array} \right) \left( \begin{array}{c} \dfrac{S \mid S}{(DP\backslash S)/DP} \\ \text{loves} \end{array} \quad \begin{array}{c} \dfrac{S \mid DP\triangleright S}{DP} \\ \text{everyone} \end{array} \right)$$

$$= \begin{array}{c} \dfrac{DP\triangleright S \mid DP\triangleright S}{S} \\ \text{his mother loves everyone} \end{array}$$

(Barker 2012, 29)

Note that loves is now lifted by S instead of DP▷S, otherwise we could not use the scope rule. Because this system has a left-right order bias we encounter a problem, namely the type cannot be lowered. If the system had a right-left bias we could lift loves with DP▷S and eventual lower it with the same lower rule.




(Barker 2006, 23)

Since we don't allow this rule derivation of "His$_1$ mother loves everyone$_1$" is impossible. But as the name implies there is something weak about these crossovers. They can be interpreted with some effort or forced by a pragmatic context. Barker argues that these interpretations can be accounted for if his system has a right left bias or if he allows other forms of types to be lowered. But right left bias or a different Lower couldn't be the basis as it would allow ungrammatical sentences to be derived. Barker essential concludes that since the indicated meaning can only be derived under special circumstances, the indicated meaning can only be derived in his systems with special rules.

## 4.4 Wh-movement

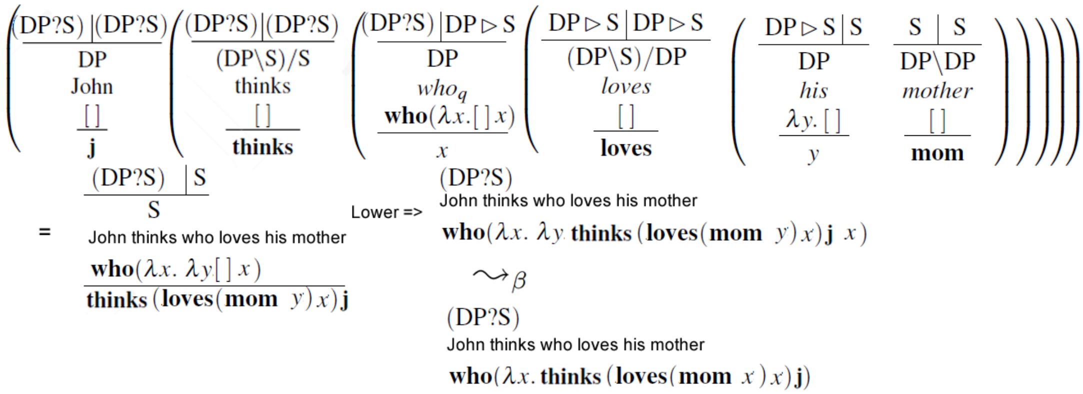
Crossover violations can also occur with wh-movement and this is where precedence differs from order evaluation because the wh-noun phrases are reconstructed at their wh-traces. Barker introduces a new connective ? as in A?B that represents an element of type B asking for type A[8]. The wh-in-situ gets almost the same type as a pronoun, difference is DP?S instead of DP$\triangleright$S. Its type is DP?S ⫽(DP⑊S) and semantically it is $\lambda K.wh(\lambda x.K\ x)$ where wh is the wh being used. In case of *who* it would be $\lambda K.who(\lambda x.K\ x)$. For wh-ex-situ Barker introduces the Gap and a type shifter called front that shifts an in-situ wh-phrase to an ex-situ wh-phrase. A Gap will be syntactical and semantical an identity function and is thus of type A⑊A and λK.K. The front type shifter will be of A ⫽B to A/B, if applied to the type of an in-situ wh-phrase we could get DP?S/(DP⑊S) , an ex-situ wh-phrase. So the ex-situ wh-phrase is not surrounded by something that is missing a DP but it's on the left of something that is missing an DP.

Strangely Barker(2012) doesn't show examples of his system with basic wh-movement with and without crossover violations. He only treats complex wh-sentences with multiple continuous levels to show how robust his system is. So lets consider a simple sentence containing a wh-phrase that binds a pronoun without a crossover violation to show how this works:

[8]in case of "who is running at great speed ?" it would be of type DP?S, the sequence is of type S and a DP is being asked, in this case the answer is Usain Bolt

(42)John thinks $who_1$ loves $his_1$ mother?

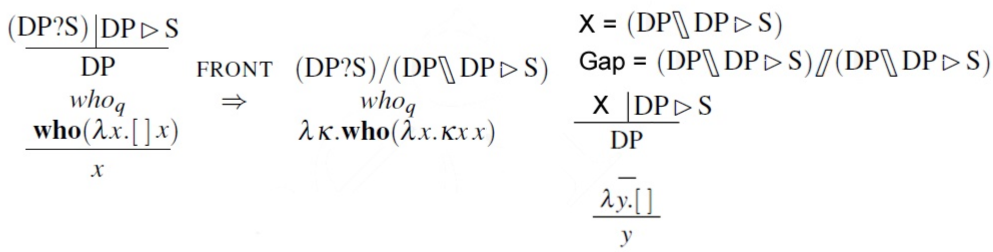
In (42) “his mother” will be the normal pronoun type DP▷S ⫽(DP⧹S) and who will be of type DP?S ⫽(DP⧹S) but in order for who to bind “his mother” we use the type shifter bind resulting in the type DP?S ⫽((DP▷S)⧹S). We use UP on loves of type (DP\S)/DP with DP▷S so the continuations match and there is a chain of DP▷S from the binder to the bindee. John of type DP and thinks of type (DP\S) will be lifted with DP?S. Combining all these elements results in (DP?S) ⫽(S⧹S) which can be lowered because it has it’s required form. I have created the derivation in tower-notion below. The end result can then be reduced with beta reduction to who (λx.thinks ( loves (mother x) x) j).

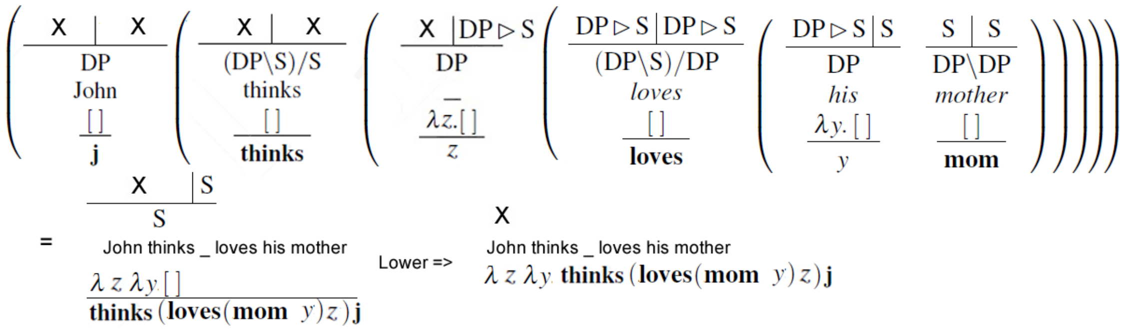


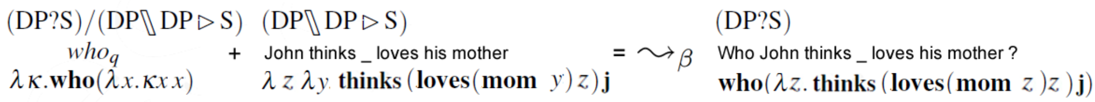

A similar procedure can be followed with the ex-situ equivalent:

(13) $Who_1$ did John think $t_1$ loves $his_1$ mom?

First we use the type shifter front on who from the last sentence. We get (DP?S)/(DP⧹(DP▷S)) which becomes it’s desired type when left of it is something of (DP⧹(DP▷S)). Informally it wants something that is a sentence with an unbounded pronoun that is missing a DP. For our convenience let’s call this call this X. Since Gap is an identity function and we want the ”John thinks _ loves his mother” to be of type X, Gap will be of Type X ⫽ X. With this in mind we can easily construct the other types. Loves, his and mother will be of the same type as the previous example but now we are lifting John and thinks with X. The end result is X ⫽(S⧹S) which can be easily lowered down to X. Now we can combine ex-situ who and X with a normal application combinator CCG get DP?S with the desired equivalent semantics as the in-situ sentences.






Now lets look at an example where there is a crossover violation. We will repeat sentence (12) from the previous chapter:

(43) $\text{He}_1$ thinks $\text{who}_1$ is tall ?
(12) $\text{Who}_1$ did $\text{he}_1$ think $\text{t}_1$ is tall?

In the in-situ variant of the sentence there is no way who can bind he since the system has a left-to right bias, with other words because he precedes who and there is no further movement there can't be any binding for the in-sito wh. Also the fronted Who in the ex-situ variant cannot bind because it's not in the desired form. X/Y instead of A $/\!\!/$(DP$\backslash\!\!\backslash$B), it has a single slash instead of a double slash. We can however derive a sentence where he is unbounded all together. This is also a correct interpretation of the sentence:

(44) $\text{Who}_1$ did $\text{he}_2$ think $\text{t}_1$ is tall?

With Barkers system as presented here we can account for variable binding of phrases containing QNP's and wh-movement. This is all done without c-command.

## 5 Conclusions

In this thesis I presented three systems. One from Büring which handles binding of pronouns in the traditional way with c-command. He alters binding condition A and B and removes condition C altogether in favor of his own Have Local Binding rule. This rule forces you to use a sentence with a bound pronoun when it has the same meaning as one with a R-expression. In the third chapter I presented the system of Bruening who argues against this tradition of c-command by introducing his own command with a notion of precedence. In Bruenings take on the binding conditions the sentences are processed from left to right. When a pronoun that is being processed refers to something in the active set C or local set A it must have a specific form. Just like the Have Local Binding rule these are not only structural but also pragmatic, as his condition C was based on Schlenker Minimize Restricors. You cannot refer to something in the active set C if it has the form of a referring expression and an alternative doesn't change the meaning or pragmatic effects. He also argued that the binding conditions and variable binding are two different things. In the last chapter I presented the system of Barker. Barker shows that these variable binding and their crossover can be taken care of without a notion of c-command but with precedence. He does this in a system that uses continuation. The semantic interpretation of pronouns are functions on their own context and a binder can then take the outcome as an argument and bind this pronoun.

These are adequate alternatives to c-command for the binding of pronouns as Bruening and Barker show how co-reference and variable binding can work without it. Bruening even makes a good case against c-command. Using a form of precedence is justified because not only does it work, it can also explain asymmetry which c-command can't. The way we naturally construct or utter sentences is also a justification to use a form of precedence. But Barker doesn't plea against c-command, he even states that to differentiate between weak crossovers and strong crossovers one maybe needs to introduce an notion of c-command (Barker 2012, p. 30). In the same sense this thesis doesn't plea against c-command but there are cerntenly flaws with it and there is an alternative without c-command that can work. These alternatives also show that precedence is of great importance.

## 6 Bibliography


Chris Barker and Shan, Chung-chieh. *Explaining crossover and superiority as left-to-right evaluation.* Linguistics and Philosophy 29.1 (2006): 91-134.

Chris Barker and Shan, Chung-chieh. *Continuations and Natural Language.* New York: Linguistics department. (2012).

Bruening, Benjamin. *Precede-and-command revisited.* Language 90.2 (2014): 342-388.

Bürings, Daniel. *Binding Theory.* Cambridge: Cambridge University Press. (2005).

Chomsky, Noam.*Lectures on Government and Binding.* Dordrecht: Foris Publications. (1981).

Haegeman, Liliane. Introduction to government and binding theory. (1991).

Reinhart, Tanya Miriam. *The syntactic domain of anaphora.* Diss. Massachusetts Institute of Technology. (1976).

Schlenker, Philippe. *Minimize restrictors!(Notes on definite descriptions, Condition C and epithets).* Proceedings of Sinn und Bedeutung. (2004).