Proceedings of Sinn und Bedeutung 11, E. Puig-Waldmüller (ed.), Barcelona: Universitat Pompeu Fabra, pp.63-75.

SPELLING OUT QR[*]

Asaf Bachrach and Roni Katzir
MIT

{asaf,trifilij}@mit.edu

Abstract

Right-Node Raising has been recently shown to license an exceptional pattern of QR. We provide a framework in which locality is determined through the interaction of complete dominance and spellout, and in which the pattern of QR in Right-Node Raising is predicted.

1 Right Node Raising and Quantifier Raising

1.1 The puzzle

Quantifier Raising (QR) is often assumed to be clause-bound (May, 1985):

(1) John knows a man who speaks every Germanic language. ($\exists \succ \forall, *\forall \succ \exists$)

As noted by Sabbagh (2003/2006), Right-Node Raising (RNR) is an exception to this generalization, allowing an embedded quantifier to take scope over a higher quantifier even across a tensed clause:

(2) [John knows a man who speaks _], and [Mary knows a woman who wants to learn _] **every Germanic language**. ($\exists \succ \forall, \forall \succ \exists$)

This generalization must be qualified, however. As we notice elsewhere (Bachrach and Katzir, 2006), a shared quantifier cannot reverse scope with a quantifier that is above the conjunction:

(3) John told me about someone who knows [a man who speaks _], and [a woman who wants to learn _] **every Germanic language**. ($\exists \succ \forall, *\forall \succ \exists$)

The emerging generalization can be formulated as follows:

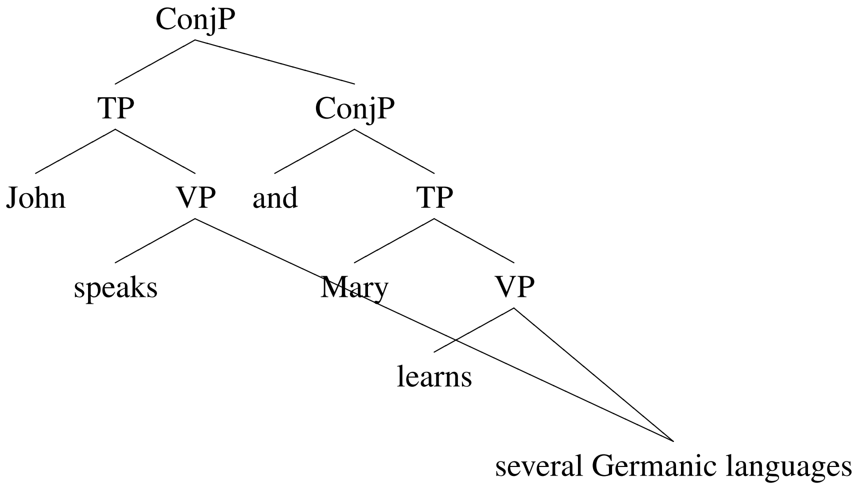
(4) **Height Generalization**: The scope of a shared quantifier (in RNR) is not clause bound within each conjunct but is clause bound above the conjunction.

The goal of this paper is to explain (4). We will adopt a phase-based architecture, based on Chomsky (2001), where certain locality effects have their source in the cyclic spellout of structures throughout the derivation. We will further assume that syntax allows sharing, or multiple-dominance, and that both movement phenomena and RNR involve multiple-dominance. In particular, we will follow McCawley (1988) and Wilder (1999) in assuming the following structure for RNR:

(5) a. John speaks and Mary learns several Germanic languages

[*]We thank Kai von Fintel, Danny Fox, Irene Heim, David Pesetsky, Yael Sharvit, Raj Singh, Benjamin Spector, and the audiences at MIT and at Sinn und Bedeutung 11 for helpful comments and discussion.

b. Structure (modeled after Wilder (1999)):



Finally, we will assume that the freezing effect of spellout applies only to material that is *completely dominated* at that point. In other words, material that is shared at the point of spellout is exempt from the locality constraints that it imposes. In particular, the sharing of a quantifier between the two conjuncts in RNR makes possible the wide-scope readings that are otherwise unavailable, but only up to the height of coordination. Once coordination is formed, the quantifier is completely dominated and is once again subject to the usual locality constraints.

The freezing effect of spellout attributes locality to the interfaces. In addition to this post-syntactic source of locality, it is commonly assumed that there are locality conditions on the operation of syntax itself (*superiority effects*), and there have been proposals to the effect that QR is a syntactic operation that is subject to these conditions. By choosing to analyze QR as a post syntactic operation we predict that any locality effects with respect to quantifier interaction would be the result of spellout and not superiority or other syntax internal constraints. In section 4 we show that the two views (the syntactic vs. post syntactic nature of QR) make different predictions, and we use further data from RNR to argue for the post-syntactic view. We will show that while QR is sensitive to spellout induced locality, it is not subject to superiority effects.

1.2 An aside: Why RNR is not movement

Before proceeding to the details of our analysis, one may wonder whether a simpler explanation to the height generalization (4) is not available. If RNR is analyzed as a form of rightward ATB movement, the shared quantifier can take a free ride on this movement and reach the height of coordination. However, there is a wide range of evidence that argues against a movement account of RNR. A full discussion of this evidence falls outside the scope of this paper (see, in particular, Abbott, 1976; Wilder, 1997; Swingle, 1995; Abels, 2004). We will only mention here that RNR violates locality conditions on movement in general and rightward movement in particular. As simple examples of this island insensitivity, (6) shows that RNR is insensitive to relative-clause islands, while (7) shows that it is insensitive to the Right-Roof Constraint. We are not familiar with movement accounts of RNR that can handle these facts.

(6) a. [John met a man who wrote _], and [Mary met a man who published _] **a recent book about bats**.

b. */? What did John meet the man who wrote a recent book about?

(7) a. * John claimed that Sam loves _ yesterday **the new headmaster**.

b. [John claims that Sam loves _], and [Joe claims that Sam hates _] **the new headmaster**.

2 Syntactic structure and the interfaces

2.1 Dominance, complete dominance, and spellout

As long as syntactic representations are trees, the notions of dominance and of complete dominance are identical: if two nodes X and Y dominate a node Z then either X dominates Y or Y dominates X. As soon as multiple dominance is allowed, however, it becomes possible for X and Y to dominate Z without either of X and Y dominating the other. That is, X and Y both dominate Z but neither *completely dominates* it. In what follows it will be important to have a precise definition of both dominance and complete dominance. We make use of a reflexive version of dominance (8), to which we add the definition of complete dominance in (9) (see Bachrach and Katzir 2006 for motivation and discussion).

(8) **Dominance:** A node X dominates Y iff (a) X is Y, or (b) a daughter of X dominates Y.

(9) **Complete Dominance:** A node X completely dominates a node Y iff (a) X is the only mother of Y, or (b) X completely dominates every mother of Y. The set of nodes completely dominated by X will be called the *Complete Dominance Domain* of X, written $CDD(X)$.

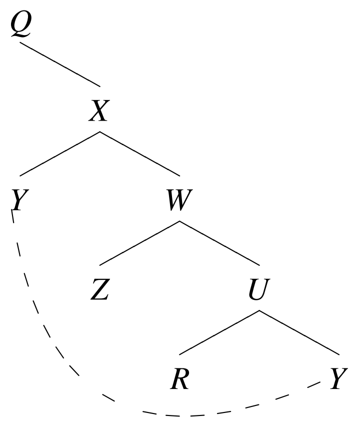
We adopt a modified version of the phase based architecture outlined in Chomsky (2001). Within this framework, the locality conditions on various grammatical operations are the result of cyclic spellout: at certain points in the derivation the structure is spelled out by being sent to the interfaces, where it is interpreted. Under the assumption that interpretation at an interface results in an immutable object, the effect of spellout is that parts of the structure are frozen at intermediate points in the derivation and cannot be further modified at later stages. If such a change is attempted, the interface will have to recompute its interpretation for the same structure, and the attempt to modify the immutable object will crash the derivation. We will refer to the points at which spellout occurs as *phase nodes*, and we will assume that what is spelled out at a phase node is everything that is *completely dominated* by that node:[1]

(10) **Phase Node:** A designated syntactic object that triggers spellout of its spellout domain.

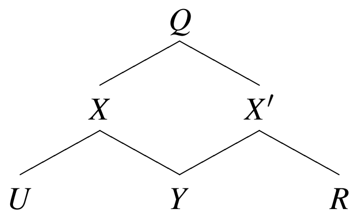
(11) **Spellout Domain:** The spellout domain of a node X is $CDD(X)$.

In example (12), which corresponds to the schematic configuration of movement, Y is merged twice: once as a daughter of U and once as a daughter of X. Notice that while X dominates Y, it does not completely dominate it according to (9). The reason is that X is neither the only mother of Y nor does it completely dominate every mother of Y (since it does not completely dominate itself). Consequently, Y is not in the spellout domain of X, and if X is a spellout node Y will not be sent to the interface at this point. This will allow Y to be re-merged again at a later stage. As long as Y keeps being re-merged as the daughter of each subsequent spellout node it will avoid being frozen by spellout. Notice that while X does not completely dominate Y, X's mother Q *does* completely dominate Y, and so Y is in its spellout domain. If Q is a spellout node, Y will be spelled out at this point.

[1]We will not try to offer a theory of phase nodes in this paper. For the purposes of this paper we will make the simplifying assumption that the set of phase nodes is identical to the set of *CP* nodes.

(12)


A different configuration that involves multiple-dominance is shown in (13). Here Y is shared by X and X'. Neither X nor X' is the only mother of Y, and neither of them completely dominates every mother of Y (since neither dominates the other). Consequently, Y is not completely dominated by either X or X', and even if one of them (or both) is a spellout node, Y will not be sent to the interface. As before, while Y is not completely dominated by X (or X'), it is completely dominated at X's mother Q.

(13)


We will assume that interpretation at the interface is a local operation applying to each node in the structure. If G is a syntactic representation then every node N in G will be mapped onto a structure at each interface: a phonological representation $P(N)$ at PF and a semantic representation $S(N)$ at LF.

2.2 Spellout at the phonetic interface

We can think of $P(N)$ as a representation of the precedence relations among the nodes dominated by N. PF imposes constraints on these relations, and in particular on the way that precedence among daughter nodes maps onto precedence at their mother. The main effect of spellout at PF is in ensuring that whatever precedence relations hold for the spellout domain of a phase node cannot be changed later on. That is, if N^* is a phase node, then $P(N^*)$ is a string composed of the elements of $CDD(N^*)$. This freezing of precedence relations provides an explanation for various locality effects in *wh*-movement. It also makes predictions about the availability of exceptional long-distance dependencies when multiple-dominance is available, as in RNR. As we have seen above, in RNR the shared material is not completely dominated in either conjunct. As a consequence, even if each conjunct contains phase nodes, the phonological position of the shared material is not frozen within either conjunct. This delay in spellout allows for the apparent violation of island constraints as in the cases of long-distance RNR (6 above, repeated below as 14), where the shared material can be pronounced to the right of the entire conjunction despite conjunct-internal phases.

(14) a. [John met a man who wrote _], and [Mary met a man who published _] **a recent book about bats**.

b. */? What did John meet the man who wrote a recent book about?

Notice that since precedence is determined syntax internally, delayed spellout does not affect it. This explains why in RNR, the shared material must be pronounced to the right of all conjuncts:

(15) * [John met a man who wrote **a recent book about bats**], and [Mary met a man who published _].

We discuss these predictions at length in Bachrach and Katzir (2006). Our main focus here is the semantic structure $S(N)$, which can be thought of as representing the hierarchical relations among the nodes dominated by N, and where multiple-dominance can also license exceptional long-distance dependencies, as we have already seen. We now turn to the details of $S(N)$.

3 Spellout at the semantic interface

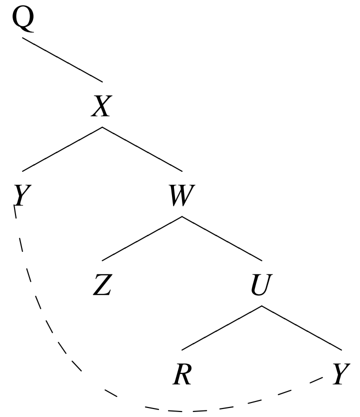
The main ingredients of $S(N)$ are the mapping from syntax to LF, the interpretation at LF, and the locality conditions imposed by spellout. We discuss each in turn.

3.1 Logical Form

3.1.1 Preliminaries: terminology for syntax

To facilitate our discussion of the mapping from syntax to logical form, we first provide a brief description of our syntactic framework. A syntactic structure is a set of nodes, where each node is identified by a unique address taken from the set $\{x_1,x_2,x_3,\ldots\}$. In addition, each node is associated with a syntactic category and with a list of addresses of its daughters. That is, each node can be described as a 3-tuple of the form *<address,category,[daughter_list]>*. For example, a syntactic structure can contain $< x_4, NP, [x_1,x_3] >$, which has x_4 as its address, NP as its category, and two daughters, one that has x_1 as its address and another that has x_3 as its address. As mentioned above, we do not allow cycles in our structures. In other words, using our earlier notion of dominance (a node dominates itself and everything dominated by its daughter), we require that no node be dominated by its daughter. There are certain other conditions on the well-formedness of syntactic structures, such as the relation between categories and selection, but we will not be concerned with those here.

Example (12) repeated here as (16a) is just a way to visualize (16b).

(16) a.


b. Nodes: $\{< x_1,R,\emptyset >, < x_2,Y,\emptyset >, < x_3,U,[x_1,x_2] >, < x_4,Z,\emptyset >, < x_5,W,[x_4,x_3] >, < x_6,X,[x_2,x_5] >, < x_7,Q,[x_6] >\}$

In what follows we will keep drawing graphs for convenience, but will keep in mind that nodes have addresses. Where no confusion arises we will also talk interchangeably about nodes and their addresses and write things like $X \in G$ to to say that X is a member of the set of nodes of G.

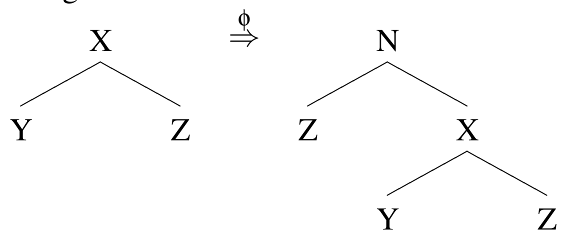
3.1.2 The mapping to Logical Form

The representation we use for logical forms will be of the same kind as the syntactic representation we have just seen. Specifically, each LF is a directed acyclic graph with nodes that have addresses taken from the set $\{x_1, x_2, x_3, \ldots\}$ and that have ordered lists of daughters. We will have nothing to say about the possible role of syntactic categories at LF. The mapping from a syntactic structure G to its logical form S is a first step in constructing semantic representations (sets of λ-expressions) from syntactic structures. Our main requirement of this mapping is that the original dominance relations remain intact:

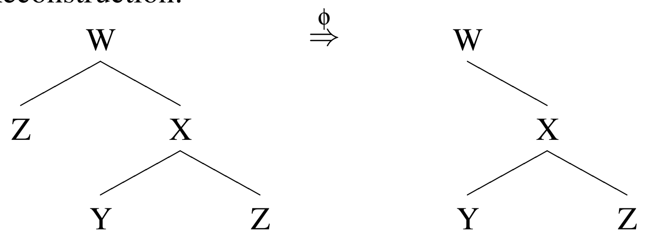
(17) Let G, S be graphs, then $\phi : G \rightarrow S$ is *dominance preserving* iff
$\forall X, Y \in G. \phi(X) \in \mathit{Dominance}_S(\phi(Y)) \leftrightarrow X \in \mathit{Dominance}_G(Y)$

Dominance preserving maps allow certain structural changes. For example, it is permitted to re-merge a node in a c-commanding position, as in the raising-like configuration in (18a), which will be the basis for our analysis of QR. It is also permitted to remove an occurrence of a node that c-commands another occurrence of itself, as in the reconstruction-like configuration in (18b). In all these cases, no hierarchical relations are created or destroyed between nodes of G. New relations are created with respect to N in (18a), but since N did not come from G, the definition of dominance preserving is satisfied.

(18) a. Raising:

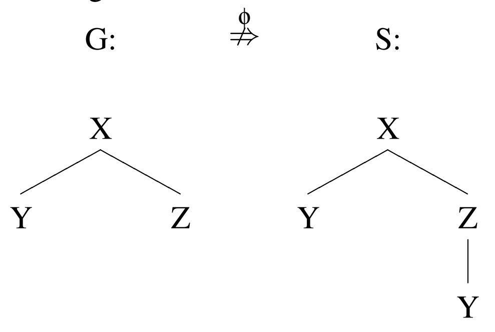


b. Reconstruction:

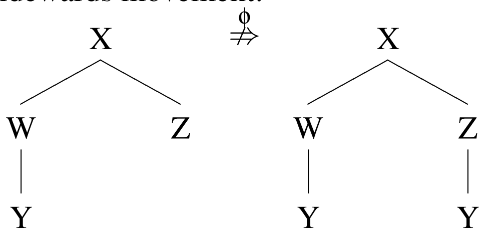


There are several configurations that dominance-preserving mappings do *not* license. We briefly mention two of them here, but a discussion of their role in interpretation lies beyond the scope of the current paper. First, re-merging a node at a lower position (19a) creates a new dominance relation and is therefore prohibited. Similarly for re-merging a node in any other position that does not c-command the original (19b). In both cases, X, Y, and Z are all in G, so the establishment of the new hierarchical relation ($Y \in \mathit{Dominance}_S(Z)$) violates order preservation.

(19) a. Lowering:



b. Sidewards movement:



3.2 Interpretation

Having constructed our LFs, we can now proceed to define the way that semantic representations are constructed. Each node $X \in S$ is mapped onto a set of semantic representations, or λ-expressions, that we will denote by $\Sigma(X)$.[2] We use sets rather than unique values in order to account for ambiguities, and more importantly because of the way in which our mechanism for variable binding will work. The elements in Σ come from two sources, the syntactic structure G and its LF S.

The syntactic structure G contributes two kinds of values: lexical entries and variables. Lexical entries are the usual kind of λ-expressions associated with words or structures, things like $\lambda z.(\textbf{walks}\ z)$ or **john**. Using φ for the mapping from G to S and $[\![\cdot]\!]$ for the mapping from structures to sets of lexical entries we can write the following rule:

(20) Lexical entries: $\forall X \in G.\forall \sigma \in [\![\text{X}]\!].\sigma \in \Sigma(\phi(X))$

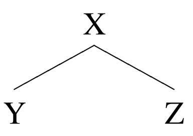
The variables that G contributes are simply the addresses of its nodes. That is, a node like $<x_{13}, NP, [x_9, x_{12}]>$ will contribute the variable x_{13} to the semantic value of its image under φ:[3]

(21) Variables: $\forall X \in G.address(X) \in \Sigma(\phi(X))$

The contributions of G are all atomic. They represent values associated with nodes in G in and of themselves without taking into account the structure in which these nodes appear. We keep compositionality, that is, the part in which the λ-expressions of smaller structures combine to form the λ-expressions of a larger one, to the logical form S. Assuming binary branching for convenience, the relevant configurations are of the form:

[2]Our concern here is in how these λ-expressions are formed. We will have nothing to say about how these λ-expressions are interpreted within a model.

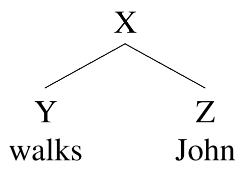
[3]For presentation purposes, we will use uppercase to denote nodes, and lowercase for addresses/variables. For example, if R is a node, r will be its address.

(22) 

The basic mode of combination for two sisters is concatenation: two semantic values, σ and τ, form the pair $(\sigma\ \tau)$. It is convenient to think of such a pair as an input to a rule like *Function Application* (Heim and Kratzer, 1998): σ is a function that takes τ as its argument. More precisely, the pair $(\sigma\ \tau)$ is the potential input to a rule of β-reduction: if σ is of the form $\lambda x.u$ then $(\sigma\ \tau)$ can be simplified to $u[\tau/x]$, the result of substituting τ for each occurrence of x in u.[4] To simple pair formation (23a) we need to add the counterpart of λ-abstraction, which, following Heim and Kratzer (1998), we condition on syntactic movement (23b).

(23) Term formation: for all $X, Y, Z \in S$, if X is the mother of Y and Z in S then
  a. If $Y \notin Dominance(Z)$, then $\forall\sigma \in \Sigma(Y).\forall\tau \in \Sigma(Z).(\sigma\ \tau) \in \Sigma(X)$
  b. If $Y \in Dominance(Z)$, and if $y = address(Y)$, then $\forall\sigma \in \Sigma(Y).\forall\tau \in \Sigma(Z).(\sigma\ \lambda y.\tau) \in \Sigma(X)$

The following examples show how semantic interpretation works in some simple cases.

(24) a. John walks
  b. 

G contributes two values for each terminal node: a lexical entry and a free variable.

(25) a. $\lambda x.(\textbf{walks}\ x) \in \Sigma(Y)$
  b. $y \in \Sigma(Y)$

(26) a. $\textbf{john} \in \Sigma(Z)$
  b. $z \in \Sigma(Z)$

For the nonterminal X, G contributes only a free variable:

(27) $x \in \Sigma(X)$

Compositionality in S contributes several values to the meaning of X: concatenating the two variables y and z (28a), the variable y with the lexical value **john** (28b), the lexical value $\lambda x.(\textbf{walks}\ x)$ with the variable z (28c), and finally the two lexical values, $\lambda x.(\textbf{walks}\ x)$ and **john** (28d).[5] From this final combination we can obtain the final representation (29) via β-reduction.

(28) a. $(y\ z) \in \Sigma(X)$
  b. $(y\ \textbf{john}) \in \Sigma(X)$
  c. $(\lambda x.(\textbf{walks}\ x)\ z) \in \Sigma(X)$
  d. ☞ $(\lambda x.(\textbf{walks}\ x)\ \textbf{john}) \in \Sigma(X)$

(29) $(\textbf{walks}\ \textbf{john}) \in \Sigma(X)$

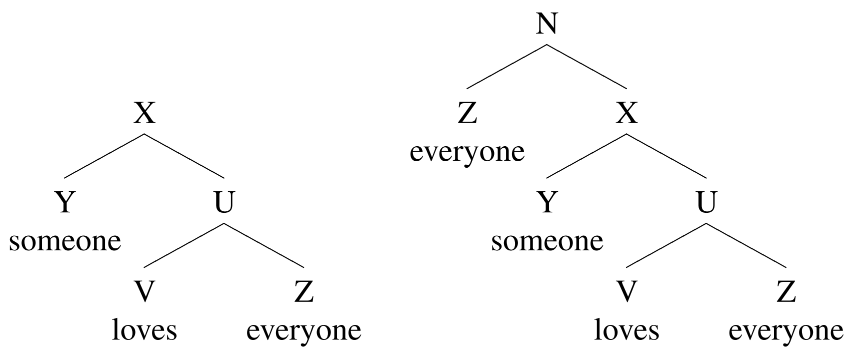
[4] Assuming that no accidental binding occurs and that the types are compatible.

[5] Here and elsewhere we use ☞to mark a semantic value that contributes to one of the final values.

To see the use of variables we will need to move to a slightly more complex example. Consider the inverse scope reading of (30). The relevant LF for this reading is the one obtained by QR (31), an operation that corresponds to the abstract configuration in (18a). As discussed above, this configuration is allowed by our dominance preserving condition (17).

(30) Someone loves everyone (*inverse scope,* $\forall \succ \exists$)

(31) G: $\overset{\phi}{\Rightarrow}$ S:



The following summarizes the contributions of G to the semantic values of its terminal nodes.

(32) a. ☞ $z \in \Sigma(Z)$
b. ☞ $\lambda f_{et}.\forall x.(f\ x) \in \Sigma(Z)$

(33) a. $v \in \Sigma(V)$
b. ☞ $\lambda x.\lambda y.((\mathbf{loves}\ x)\ y) \in \Sigma(V)$

(34) a. $y \in \Sigma(Y)$
b. ☞ $\lambda f_{et}.\exists x.(f\ x) \in \Sigma(Y)$

Compositionality in S provides the many different values $\Sigma(U)$:[6] The only one that is relevant here is the term formed by taking the lexical value of $\Sigma(V)$ (33b) and the variable of $\Sigma(Z)$ (32a):

(35) ☞ $(\lambda x.\lambda y.((\mathbf{loves}\ x)\ y)\ z) \in \Sigma(U)$

From (35) we obtain the following via β-reduction:

(36) $\lambda y.((\mathbf{loves}\ z)\ y) \in \Sigma(U)$

Along similar lines, compositionality contributes several different values to $\Sigma(X)$. The only one that is relevant here is the result of concatenating the lexical value of $\Sigma(Y)$ (34b) and (36):

(37) ☞ $(\lambda f_{et}.\exists x.(f\ x)\ \lambda y.((\mathbf{loves}\ z)\ y))$

Applying β-reduction to (37) we obtain

(38) $\exists x.(\lambda y.((\mathbf{loves}\ z)\ y)\ x) \in \Sigma(X)$

And then

(39) $\exists x.((\mathbf{loves}\ z)\ x) \in \Sigma(X)$

[6]From the lexical stage we also have $u \in \Sigma(U)$. If the opposite order is also considered, $Z\ V$, more values will result. On the other hand, type-checking may eliminate some of the potential values. None of this is important for the current discussion.

Finally, by using the lexical value for Z (32b) and the value (39) for X we obtain the following value for $\Sigma(N)$:

(40) ☞ $(\lambda f_{et}.\forall x.(f\ x)\ \lambda z.\exists x.((\textbf{loves}\ z)\ x)) \in \Sigma(N)$

From this, by simple conversion we obtain the desired interpretation (each line is derived from the previous one):

(41) a. $(\lambda f_{et}.\forall y.(f\ y)\ \lambda z.\exists x.((\textbf{loves}\ z)\ x)) \in \Sigma(N)$ (by α-conversion)
b. $(\forall y.(\lambda z.\exists x.((\textbf{loves}\ z)\ x)\ y)) \in \Sigma(N)$ (by β-reduction)
c. $\forall y.\exists x.((\textbf{loves}\ y)\ x) \in \Sigma(N)$ (by β-reduction))

3.3 Locality

The final component in our semantic framework is the locality constraint imposed by spellout. Recall from our abstract discussion above that when spellout occurs at a phase node, the spellout domain of that node is mapped onto an immutable object. Operations that occur at later stages cannot modify this object. Within the current framework, long-distance effects are mediated through variables. The only way in which a semantic value σ can be modified by a subsequent operation is if σ contains a free variable that becomes bound at a later stage. The freezing effect of spellout, then, must involve the removal of any free variables that correspond to nodes that are completely dominated by the current phase node.

(42) $\forall X \in Phase-node(G).\forall \sigma \in \Sigma(\phi(X)).\forall Y \in CDD(X).y \notin FV(\sigma)$

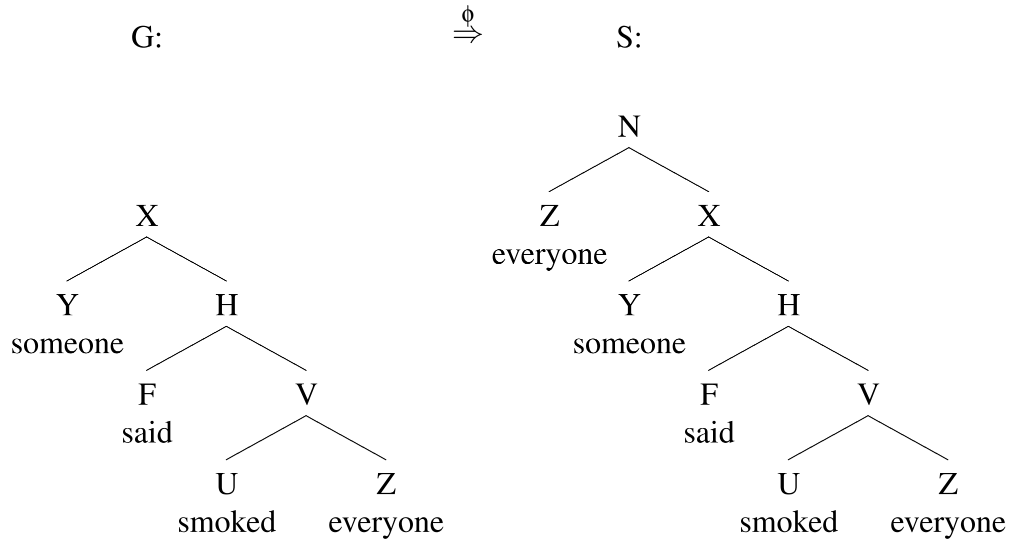
Where $FV(t)$, the free variables of a term t, are defined as:

(43) a. $FV(x) = \{x\}$
b. $FV((t_1 t_2)) = FV(t_1) \cup FV(t_2)$
c. $FV(\lambda x.t) = FV(t) \setminus \{x\}$

We have seen above that in order to obtain the scope reversal effects of QR, the lower quantifier must be interpreted twice: as a variable in its lower position, and as its lexical entry in its higher position, to which it is re-merged at LF. All the nodes between the two positions have a free variable corresponding to the quantifier. The immediate effect of the spellout condition (42), then, is that QR is impossible across phases.

(44) a. Someone said everyone smoked *(no inverse scope,* $*\forall \succ \exists$*)*

b.



Before we show why (44b) does not give rise to the required inverse-scope reading, notice that there is nothing wrong with the mapping itself. As in our discussion of (31) above, ϕ is a dominance preserving mapping, corresponding to the raising configuration in (18a). The first few steps of interpretation also proceed as before. G contributes both the variable z and the lexical value $\lambda f_{et}.\forall x.(f\ x)$ to the semantics of Z. Since $Z \in \mathit{Dominance}(X)$, the elements in $\Sigma(N)$ will be of the form $(\sigma\ \lambda z.\tau)$, where $\sigma \in \Sigma(Z), \tau \in \Sigma(X)$, or they can be reductions of such terms. The reading we are interested in is $\forall x.\exists y.((\textbf{said}\ y\ (\textbf{smoked}\ x)))$. The only potential elements in $\Sigma(N)$ that can be used to obtain this reading are derived from:

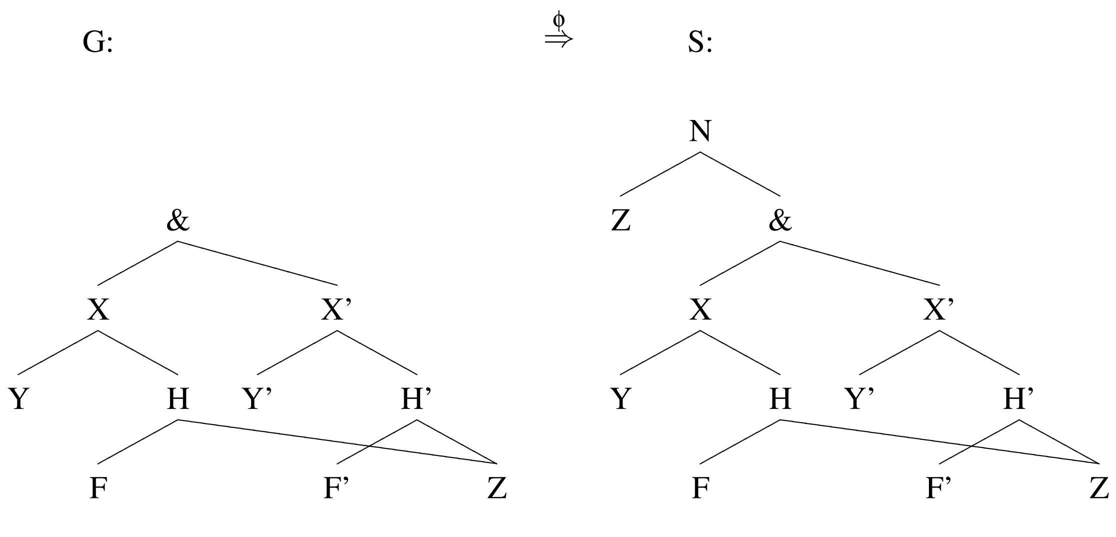
(45) $(\lambda f_{et}.\forall x.(f\ x)\ \lambda z.\exists y.((\textbf{said}\ y\ (\textbf{smoked}\ x))))$

This, in turn, means that $\Sigma(X)$ contains

(46) $\exists y.((\textbf{said}\ y\ (\textbf{smoked}\ x))))$

Notice, however, that in our sentence the two quantifiers are separated by at least one phase node: the *CP* node of the embedded clause. This means that spellout has occurred somewhere below X (say, at V). At that phase Z was completely dominated, so z cannot occur as a free variable in any meaning of that phase. In particular, $\exists y.((\textbf{said}\ y\ (\textbf{smoked}\ x)))$ is filtered out. Consequently the inverse scope reading cannot be obtained in N.

As discussed above, spellout freezes only those nodes that are *completely* dominated by a phase node. This suggests an escape hatch to the clause-boundedness of QR. If the lower quantifier can be shared at the point of intermediate phase nodes it will be excluded from the complete dominance domain of those nodes and will consequently be able to remain interpreted as a free variable. Later on, QR will enable it to bind this variable, resulting in scope reversal. As we have already seen, RNR provides precisely this option. Here is how it works.

(47)


The shared material Z is not completely dominated inside either conjunct. Consequently, the variable z can remain free despite conjunct-internal spellout. After conjunction, raising of the shared material (in S) will bind the variable z, resulting in scope reversal.

4 Does QR obey superiority?

The picture of QR that emerges from the discussion above is as a post-syntactic operation. While we have attempted to maintain as much as possible from the insights of the view of QR as movement, the operation we end up with differs from familiar forms of A' movement in being freely available (and limited only by interface conditions) and in not feeding back into syntax. A very different view of QR has been advocated recently by Bruening (2001), based on scope interactions in double-object constructions. As has been known since Larson (1985), scope reversal is not available between the two objects in a double-object construction, though they are possible in the corresponding dative structure.

(48) a. I gave a child every doll ($\exists \succ \forall, *\forall \succ \exists$)
b. I gave a doll to every child ($\exists \succ \forall, \forall \succ \exists$)

Bruening observes that while the relative scope of the two objects is frozen, the scope of the second argument is not frozen with respect to other scope bearing elements in the sentence, as, for example, the subject in 49:

(49) A (different) student gave me every manuscript ($\exists \succ \forall, \forall \succ \exists$)

Based on these data, Bruening argues for a feature-driven notion of QR, in which the raised quantifier is sensitive to the principle of *Attract Closest*, a distance-sensitive economy condition of the kind that is often assumed to govern *wh*-movement (cf. Richards, 1997). If QR is feature driven, then attracting the second object across the first in the case that both are quantifiers will violate *Attract Closest*.[7] The only way in which the second quantifier can outscope the subject is if both objects move above the subject.

The feature-driven view of QR thus offers an elegant explanation to the scope interactions in double-object constructions. Nevertheless, we believe that this view cannot be maintained in the face of the RNR facts. In addition to the arguments that we have already discussed for a post-syntactic notion of QR, RNR offers a more direct argument against a feature-driven account. While RNR bleeds locality effects that originate from spellout, it does *not* bleed locality effects that originate from Relativized Minimality effects:

(50) * What$_i$ did who write a book about t$_i$?

(51) * What$_i$ did [who write _] and [who edit _] **a book about t$_i$**?

Under the current analysis, this fact is not surprising. RNR does not involve movement, and so the subject *wh*-elements in each conjunct remain as interveners when *wh*-movement attempts to attract the shared *wh*-object. The current analysis makes a further prediction. Since we analyze QR as a freely occurring operation that is frozen only through the interaction of spellout and complete dominance, we predict that RNR of the second object in a double-object construction will allow it to raise and outscope the first object. Under a feature-driven account of QR, on the other hand, the same intervention would be expected for QR as we have just seen in (51) for *wh*-movement. As far as we can determine from our informants, RNR does allow scope reversal in double-object constructions, thus providing evidence against the feature-driven account:

(52) [Mary promised some student _] but [ended up giving some professor _] **every journal in her collection.** ($\exists \succ \forall, \forall \succ \exists$)

References

Abbott, Barbara. 1976. Right Node Raising as a test for constituenthood. *Linguistic Inquiry* 7:639–642.

[7]Under the common assumption that the first object in a double-object construction c-commands the second object (Barss and Lasnik, 1986).

Abels, Klaus. 2004. Right node raising: Ellipsis or across the board movement? In *Proceedings of NELS 34*, ed. Keir Moulton and Matthew Wolf. Amherst, MA.

Bachrach, Asaf, and Roni Katzir. 2006. Right-node raising and delayed spellout. In *Proceedings of Interphases*.

Barss, Andrew, and Howard Lasnik. 1986. A note on anaphora and double objects. *Linguistic Inquiry* 17:347 – 354.

Bruening, Benjamin. 2001. QR obeys superiority: Frozen scope and ACD. *Linguistic Inquiry* 32:233–273.

Chomsky, Noam. 2001. Derivation by phase. In *Ken Hale: a life in language*, ed. Michael Kenstowicz, 1–52. MIT Press.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in generative grammar*. Oxford: Blackwell.

Larson, Richard. 1985. Quantifying into np. Ms., MIT.

May, Robert. 1985. *Logical form: Its structure and derivation*. Cambridge, Mass.: MIT Press.

McCawley, James D. 1988. *The syntactic phenomena of English*. Chicago: The University of Chicago Press.

Richards, Norvin. 1997. What moves where when in which language? Doctoral Dissertation, MIT.

Sabbagh, Joseph. 2003/2006. Ordering and linearizing rightward movement. To appear in Natural Language and Linguistic Theory.

Swingle, Kari. 1995. On the prosody and syntax of Right Node Raising. Ms., UCSC.

Wilder, Chris. 1997. Some properties of ellipsis in coordination. In *Studies on Universal Grammar and Typological Variation*, ed. Artemis Alexiadou and Tracy Hall, 59–107. Amsterdam: Benjamins.

Wilder, Chris. 1999. Right-Node Raising and the LCA. In *Proceedings of WCCFL 18*, ed. Sonya Bird, Andrew Carnie, Jason D. Haugen, and Peter Norquest.