Generalized quantifiers and their elements: operators and their scopes

2.1 Generalized quantifiers – heroes or old fogeys?

Starting with Montague (1974a) but at least with the almost simultaneous appearance of Barwise and Cooper (1981), Higginbotham and May (1981), and Keenan and Stavi (1986) generalized quantifiers became the staple of formal semantics. For decades it has been taken for granted that they serve as the interpretations of the most widely researched grammatical category in the field, i.e. noun phrases. Nevertheless, there is mounting evidence that generalized quantifiers are not the panacea magna they were once thought to be, and these days one reads more about what they cannot do than about what they can. So are generalized quantifiers a thing of the past? If not, what are they good for? What are the main reasons for them to be superseded, and by what?

Like many other books, this one starts out with generalized quantifiers, but it does so bearing the controversy around them in mind. This will also make it easier to highlight some of the underlying assumptions and some of the firm advantages of generalized quantifiers. Building on these foundations the book will survey two areas of research. One has to do with alternative approaches to scope assignment. The other has to do with the diversity in the behavior of quantifier phrases and with recent attempts to explain it in a compositional fashion. In this way the book will place an emphasis on ongoing work. Apart from the hope of stimulating research in these newer areas, making the unquestioningly generalized-quantifier-theoretic part relatively brief is justified by the fact that there are so many superb texts available on the topic. From the 1990s one would recommend Keenan (1996), Keenan and Westerståhl (1997), and Landman (1991). In recent years the most comprehensive and authoritative text is Peters and Westerståhl (2006); Glanzberg (2006) and Ruys and Winter (2008) are excellent handbook chapters.
2.2 Generalized quantifiers and their elements: operators and their scopes

In many logics, operators are introduced syncategorematically. They are not expressions of the logical language; the syntax only specifies how they combine with expressions to yield new expressions, and the semantics specifies what their effect is:

(1) If $\phi$ is a formula, $\forall x[\phi]$ is a formula. $\forall x[\phi]$ is true if and only if every assignment of values to the variable $x$ makes $\phi$ true.

The quantifier prefix $\forall x$ functions like a diacritic in the phonetic alphabet: ‘ is not a character of the IPA but attaching it to a consonant symbol indicates that the sound is palatal (e.g. [t’]). In line with most of the linguistic literature we are going to assume that operators embodied by morphemes or phrases are never syncategorematic. But if every and every dragon are ordinary expressions that belong to some syntactic category, then, by the principle of compositionality, they must have their own self-contained interpretations. This contrasts with the situation in predicate logic. In (2) the contributions of every and every dragon are scattered all over the formula without being subexpressions of it. Everything in (2) other than guard treasure’ comes from every dragon, and everything other than guard treasure’ and dragon’ comes from every.

(2) Every dragon guards treasure.
$\forall x[\text{dragon’}(x) \rightarrow \text{guard treasure’}(x)]$

Not only would we like to assign a self-contained interpretation to every dragon, we would also like to assign it one that resembles, in significant respects, the kind of interpretations we assign to Smaug and more than three dragons. The reason why these are all categorized as DPs in syntax is that they exhibit very similar syntactic behavior. It is then natural to expect them to have in some respects similar semantics. If they did not, then the syntactic operations involving DPs (e.g. merging DP with a head, in current terminology) could not be given uniform interpretations. To a certain point it is easy to see how that interpretation would go. Assume that the DP Smaug refers to the individual $s$ and the predicate (TP, a projection of Tense) guards treasure to the set of individuals that guard treasure. Interpreting the DP–TP relation as the set theoretical element-of relation, Smaug guards treasure will be interpreted as $s \in \text{guard treasure’}$. Now consider Every dragon guards treasure. The DP every dragon does not denote an individual, but we can associate with it a unique set of individuals, the set of dragons. Reinterpreting DP–TP using the subset relation, Every dragon guards treasure is compositionally
interpreted as $dragon' \subseteq guard\ treasure'$. To achieve uniformity, we can go back and recast $s \in guard\ treasure'$ as $\{s\} \subseteq guard\ treasure'$, with $\{s\}$ the singleton set that contains just Smaug. But indefinite DPs like *more than three dragons* still cannot be accommodated, because there is no unique set of individuals they could be associated with. In a universe of just 5 dragons, sets of more than three dragons can be picked in various different ways.

One of Montague’s (1974a) most important innovations was to provide a self-contained and uniform kind of denotation for all DPs in the form of generalized quantifiers, introduced mathematically in Mostowski (1957) based on Frege’s fundamental idea. The name is due to the fact we generalize from the first order logical $\forall$ and $\exists$ and their direct descendants *every dragon* and *some dragon* to the whole gamut, *less than five dragons*, *at least one dragon*, *more dragons than serpents*, the *dragon*, etc., even including proper names like *Smaug*.

A generalized quantifier is a set of properties. In the examples below the generalized quantifiers are defined using English and, equivalently, in the language of set theory and in a simplified Montagovian notation, to highlight the fact that they do not have an inherent connection to any particular logical notation. The main simplification is that we present denotations extensionally. Thus each property is traded for the set of individuals that have the property (rather than the intensional analogue, a function from worlds to such sets of individuals), but the term “property” is retained, as customary, to evoke the relevant intuition. This approach fits all three of our examples equally well:

(3) a. *Smaug* denotes the set of properties that Smaug has. If Smaug is hungry, then the property of being hungry is an element of this set.
   b. *Smaug* denotes $\{P : s \in P\}$. If Smaug is hungry, then $\{a : a \in hungry'\} \in \{P : s \in P\}$.
   c. *Smaug* denotes $\lambda P[P(s)]$. If Smaug is hungry, then $\lambda P[P(s)](hungry')$ yields the value True.

(4) a. *Every dragon* denotes the set of properties that every dragon has. If every dragon is hungry, then the property of being hungry is an element of this set.
   b. *Every dragon* denotes $\{P : dragon' \subseteq P\}$. If every dragon is hungry, then $\{a : a \in hungry'\} \in \{P : dragon' \subseteq P\}$.
   c. *Every dragon* denotes $\lambda P\forall x[dragon'(x) \rightarrow P(x)]$. If every dragon is hungry, then $\lambda P\forall x[dragon'(x) \rightarrow P(x)](hungry')$ yields the value True.
(5) a. *More than one dragon* denotes the set of properties that more than one dragon has. If more than one dragon is hungry, then the property of being hungry is an element of this set.

b. *More than one dragon* denotes \( \{ P : |dragon \cap P| > 1 \} \). If more than one dragon is hungry, then \( \{ a : a \in \text{hungry}' \} \in \{ P : |dragon \cap P| > 1 \} \).

c. *More than one dragon* denotes \( \lambda P \exists x \exists y [x \neq y \land \text{dragon}' (x) \land \text{dragon}' (y) \land P(x) \land P(y)] \). If more than one dragon is hungry, then \( \lambda P \exists x \exists y [x \neq y \land \text{dragon}' (x) \land \text{dragon}' (y) \land P(x) \land P(y)](\text{hungry}') \) yields the value True.

To make this set of sets of individuals more vivid, it is useful to invoke some simple notions of set theory. The powerset of a set \( A \) is the set of all \( A \)'s subsets. The powerset is so called because a set of \( n \) elements has \( 2^n \) subsets (2 to the \( n \)th power). Imagine a universe of discourse with 4 elements. Its powerset, i.e. the set of all its 16 subsets, is as follows:

\[
\begin{align*}
&\{\emptyset\}, \{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \\
&\{a,c\}, \{a,d\}, \{b,c\}, \{b,d\}, \{c,d\}, \{a,b,c\}, \{a,b,d\}, \{a,c,d\}, \\
&\{b,c,d\}, \{a,b,c,d\}.
\end{align*}
\]

Extensional semantics can distinguish just these 16 sets of individuals (properties) in a 4-element universe. For example, if the set of dragons is \( \{a, b, c\} \) and the set of things that fly is \( \{a, b, d\} \), then the properties of being a dragon and being a thing that flies can be distinguished. But if both sets happen to have the same elements, then an extensional semantics cannot distinguish them.

Some sets in the universe have names such as *dragon, flies, etc.* whereas others do not. But for our purposes all these are on a par. The most useful label for \( \{a, b\} \) is not ‘dragon that flies’ but, rather, ‘entity that is identical to \( a \) or \( b \)’. When we ask whether a particular sentence, e.g. *Smaug flies* is true, we are interested in sets with particular linguistic labels, but when we study the quantifiers themselves, we are interested in all the sets that are elements of the quantifier and in their relation to all the other subsets of the universe.

To visualize a generalized quantifier we draw the Hasse-diagram of the powerset of the universe. The lines represent the subset relation, thus \( \{a\} \) is below \( \{a,b\} \) and \( \{a,b\} \) below \( \{a,b,c\} \), because \( \{a\} \subseteq \{a,b\} \subseteq \{a,b,c\} \). Each generalized quantifier is represented as an area (a subset) in this diagram. If Smaug is the individual \( a \), and the set of dragons is \( \{a,b,c\} \), the generalized quantifiers denoted by the DPs *Smaug, every dragon, and more than one dragon* are the shaded areas in Figures 2.1, 2.2, and 2.3, respectively. Such diagrams will be used over and over in Chapter 4.
2.2 Generalized quantifiers and their elements: operators and their scopes

\[
\{a,b,c,d\} \quad \{a,b,d\} \quad \{a,c,d\} \quad \{b,c,d\} \quad \{a,b\} \quad \{a,c\} \quad \{a,d\} \quad \{b,c\} \quad \{b,d\} \quad \{c,d\} \\
\{a\} \quad \{b\} \quad \{c\} \quad \{d\} \quad \emptyset
\]

Fig. 2.1 The set of properties Smaug has: all the sets that have \(a\) as an element

\[
\{a,b,c,d\} \quad \{a,b,d\} \quad \{a,c,d\} \quad \{b,c,d\} \quad \{a,b\} \quad \{a,c\} \quad \{a,d\} \quad \{b,c\} \quad \{b,d\} \quad \{c,d\} \\
\{a\} \quad \{b\} \quad \{c\} \quad \{d\} \quad \emptyset
\]

Fig. 2.2 The set of properties every dragon has: all the sets that have \(\{a, b, c\}\) as a subset

\[
\{a,b,c,d\} \quad \{a,b,d\} \quad \{a,c,d\} \quad \{b,c,d\} \quad \{a,b\} \quad \{a,c\} \quad \{a,d\} \quad \{b,c\} \quad \{b,d\} \quad \{c,d\} \\
\{a\} \quad \{b\} \quad \{c\} \quad \{d\} \quad \emptyset
\]

Fig. 2.3 The set of properties more than one dragon has: all the sets whose intersection with \(\{a, b, c\}\) has more than one element
Recall that our desire for a uniform interpretation stems from the fact that all DPs play similar roles in syntax. We now have such an interpretation. The specific notion of a generalized quantifier is furthermore useful in two main respects. First, it provides a foundation for the treatment of quantifier scope. Second, it enables one to study the semantic properties of DPs, and to do so in a way that possibly subsumes them under cross-categorial generalizations. We start with scope. The property *(is) hungry* mentioned above has a simple description, but that is an accident. Properties might have arbitrarily complex descriptions:

(7) If every dragon flies or lumbers, then the property of being an individual such that he/she/it flies or he/she/it lumbers is in the set of properties every dragon has.

(8) If there is more than one dragon that spotted every adventurer, then the property of being an individual such that he/she/it spotted every adventurer is an element of the set of properties more than one dragon has.

(9) If every adventurer was spotted by more than one dragon, then the property of being an individual such that there is more than one dragon that spotted him/her/it is an element of the set of properties every adventurer has.

Properties with simple descriptions and ones with complex descriptions are entirely on a par. We are not adding anything to the idea of generalized quantifiers by allowing properties of the latter kind. But once the possibility is recognized, quantifier scope is taken care of. In each case above, some operation is buried in the description of the property that is asserted to be an element of the generalized quantifier. In (7) the buried operation is disjunction; thus (7) describes a configuration in which universal quantification scopes over disjunction. (8) and (9) correspond to the subject wide scope, S > O, and the object wide scope, O > S, readings of the sentence *More than one dragon spotted every adventurer*. In (8) the main assertion is about the properties shared by more than one dragon, thus the existential quantifier in subject position is taking wide scope. In (9) the main assertion is about the properties shared by every man, thus the universal quantifier in object position is taking wide scope.

This is all there is to it:

(10) **Scope**
    The scope of a quantificational DP, on a given analysis of the sentence, is that part of the sentence which denotes a property that is asserted to be an element of the generalized quantifier denoted by DP on that analysis.
2.3 Scope and constituent structure

2.3.1 The basic idea

The scope of an operator in logic is simply the constituent that it is attached to. All properties of absolute and relative scope follow from this.

In talking about natural language one has to distinguish between semantic scope, as in (10), and syntactic domain. In her pioneering and immensely influential work on syntactic domains for semantic rules, Reinhart (1979, 1983) hypothesized the following:

(11) Hypothesis about Scope and Domain
    The semantic scope of a linguistic operator coincides with its domain in some syntactic representation that the operator is part of.

Reinhart defines the syntactic domain of an expression as its sister relative to the first branching node above it (the expression c-commands the nodes in its sister). Her specific assumption in these works is that the only relevant syntactic representation is surface structure, but the key idea is the more general one, namely, that syntactic structure determines semantic scope and does so in a very particular way. This is not the only possible view: for example, Cooper (1983) and Farkas (1997a) put forth non-structural theories of scope. So one important task for work on the syntax/semantics interface is to determine whether (11) is correct, and if yes, exactly what kind of syntactic representations and notion of domain bear it out.

On the structural view of scope the readings in (7), (8), and (9) correspond to the semantic constituent structures (12), (13), and (14), respectively:

(12) (Every dragon) ((flies) or (lumbers))
(13) (More than one dragon) ((spotted) (every adventurer))
(14) ((More than one dragon) (spotted)) (every adventurer)

How well do these semantic constituents match up with syntactic constituents? Initial encouragement comes from the fact that overt wh-fronting creates similar constituents. *Flies or lumbers, spotted every adventurer*, and *more than one dragon spotted* are semantic constituents in (12)–(13)–(14) and syntactic constituents in (15)–(16)–(17).

(15) Who flies or lumbers?
12 Generalized quantifiers and their elements: operators and their scopes

(16) Who spotted every adventurer?
(17) Who did more than one dragon spot?

So such constituents are syntactically possible; but the question remains as to whether all scope-semantically motivated constituents are syntactically plausible.

In this section we consider two rather different ways to implement the above ideas concerning scope and to answer these questions. The approaches of Montague and May produce the above constituent structures in abstract syntax, whether or not there is independent purely syntactic evidence for them. In contrast, the approaches of Hendriks and of Barker and Shan dissociate scope from pure syntax. Their systems allow one to maintain whatever constituent structure seems motivated on independent syntactic grounds and still deliver all imaginable scope relations. Finally, the proof-theoretical perspective in Jäger (2005) and Barker (2007) offers a way to move between these as desired.

The goals of this discussion are twofold. One is to introduce some fundamental technologies. Another is to show that there is no deep semantic necessity to opt for one technology or the other; the choices can be tailored to what one finds insightful and what the empirical considerations dictate.

2.3.2 The (first) proper treatment of quantification: Montague

We consider two derivations of More than one dragon spotted every man in an extensionalized version of Montague’s PTQ (1974a). Montague used a syntax inspired by but not identical to a categorial grammar and built sentences bottom-up. This was very unusual at the time when linguists used top-down phrase structure rules, but today, in the era of Merge in Minimalism, it should look entirely natural.

We assume verbs to denote functions of individuals (entities of type e).4 Because quantifier phrases do not denote individuals, they cannot serve as arguments of such verbs. In line with the reasoning above, quantifier phrases combine with expressions that denote properties, and the semantic effect of the combination is to assert that the property is an element of the generalized quantifier. The subject being the highest, i.e. last, argument of the verb, inflected verb phrases denote a property anyway, so a subject quantifier phrase can enter the sentence without further ado. If the quantifier phrase is not the last argument, the derivation must ensure that a property-denoting expression is formed in one way or another. This is what Heim and Kratzer (1998: Chapter 7) refer to as Quantifier Raising forced by a type-mismatch.

Montague’s PTQ offers several ways to build the subject wide scope, S > O, and the object wide scope, O > S, readings of a sentence. Those
chosen below will make the relation between Montague’s, May’s, and Hendriks’s methods the most transparent. We start by applying the verb to arguments interpreted as free individual variables and build a sentence without quantifiers. Montague notated such variables with indexed pronouns in the syntactic derivation; we employ indexed empty categories $ec$.

Properties (of type $⟨e, t⟩$) are then formed from this sentence by abstracting over the variables one by one. Abstraction is achieved by binding the variable by a $\lambda$-operator.

(18) If $\alpha$ is an expression, $\lambda x[\alpha]$ is an expression. $\lambda x[\alpha]$ denotes a function of type $⟨b, a⟩$, where $b$ is the type of the variable $x$ and $a$ is the type of the function value $\alpha$. When applied to some argument $\beta$ of the same type as $x$, the value of the function is computed by replacing every occurrence of $x$ bound by $\lambda x$ in $\alpha$ by $\beta$. E.g. $\lambda x[x^2](3) = 3^2$.

Each time a property is formed, a quantifier can be introduced. The later a quantifier is introduced, the wider its scope: other operators may already be buried in the definition of the property that it combines with, and therefore they fall within its scope. The derivation of the reading where the subject existentialscopes over the direct-object universal is given first. Recall that it is to be read bottom-up, starting with “build a sentence with two free variables.” The cardinality quantifier more than one will be abbreviated using $\exists >1$. The last step is spelled out in (20).

(19) Subject $>$ Object reading

\[
\begin{align*}
\lambda P \exists_{>1} z [dragon'(z) \land P(z)](\lambda x_2 \forall y[man'(y) \rightarrow spot'(y)(x_2)]) &= \\
\exists_{>1} z [dragon'(z) \land \lambda x_2 \forall y[man'(y) \rightarrow spot'(y)(x_2)](z)] &= \\
\exists_{>1} z [dragon'(z) \land \forall y[man'(y) \rightarrow spot'(y)(z)]] &= 
\end{align*}
\]
The derivation of the reading where the direct-object universal scopes over the subject existential differs from the above in just one respect: properties are formed by \( \lambda \)-binding the subject variable first and the direct-object variable second, which reverses the order of introducing the two quantifier phrases. The last step that introduces the universal is in (22):

\[
(21) \quad \text{Object } \succ \text{Subject reading}
\]

\[
\lambda Q \forall y[\text{man}'(y) \rightarrow \exists z[\text{dragon}'(z) \land \text{spot}'(y)(z)]]
\]

Montague's PTQ collapsed the two steps of \( \lambda \)-binding a free variable and applying a generalized quantifier to the property so formed into a single rule of quantifying-in. To make the derivation more transparent, we disentangled the two steps, as do Heim and Kratzer (1998), who construe \( \lambda \)-abstraction as the reflex of the movement of the index on the variable. We followed PTQ in replacing the variable with the quantifier phrase in the surface string. This feature is syntactically unsophisticated and need not be taken too seriously; see May and Hendriks below.

Before turning to other ways to achieve the same interpretive results we take a brief look at quantifiers binding pronouns, among other reasons in order to explain why it makes sense for this book to set them aside.

### 2.3.3 Interlude: quantifier phrases do not directly bind pronouns

Predicate-logical quantifiers do not only bind variables that allow them to function as arguments of predicates, (23), which can be seen to translate one reading of (24), contains three bound occurrences of the variable \( x \), of which the one in \( \text{room-of}''(x) \) corresponds to the pronoun his.
2.3 Scope and constituent structure

\(\forall x [\text{boy}(x) \rightarrow \text{in-room-of}'(x)(x)]\)

(24) Every boy is in his room.

Is the relation between every boy and his a case of binding in the same sense as the relation between \(\forall x\) and the \(x\) of room-of\('(x)\) is, as has often been assumed? Nothing in our account of quantifier phrases as generalized quantifiers explains how they bind pronouns!

This is in fact as it should be. The bound reading of the pronoun in (24) does not come about in the same way as the binding of the \(x\)'s in (23). In (23) the three variables are all directly bound by \(\forall x\) because, in addition to being within its scope, they happen to have the same letter as the quantifier prefix. In contrast, pronouns are not directly bound by quantifier phrases in natural language. In the well-known parlance of syntactic Binding Theory, pronouns have to be co-indexed with a c-commanding item in argument position (subject, object, possessor, etc.), not with one in operator position (the landing site of wh-movement or the adjoined position created by Quantifier Raising). The claim that syntactic binding is a relation between argument positions is grounded primarily in data about reflexives but it is thought to extend to pronouns and offers a simple account of strong and weak crossover.

If the pronoun is directly linked to the c-commanding argument position and not to the quantifier itself, what is the actual operator that binds it? It is the operator that identifies the pronoun with a c-commanding argument position. The technologies for achieving identification are varied, but the interpretive result is always the same. (25) presents three equivalent metalinguistic descriptions of the bound pronoun reading of the VP saw his/her/its own father:

(25) a. be an individual such that he/she/it saw his/her/its own father
   b. \(\{ a : a \text{ saw a's father}\} \)
   c. \(\lambda x [x \text{ saw x's father}] \)

So the operator that binds the pronoun is the abstraction operator \(\lambda\). (Montague’s original grammar in PTQ makes the binding of pronouns part of the job of his rule of quantification. The reason is that, as was mentioned, he collapses predicate abstraction and applying the generalized quantifier to the predicate into a single rule. Decoupling the two, as in Heim and Kratzer (1998) and in the discussion above, is motivated not only by notational transparency but also by the reasoning in this section.)

The property in (25) combines with a noun phrase denotation as other properties do, and the pronoun’s antecedent is specified:
16 Generalized quantifiers and their elements: operators and their scopes

(26) If every girl saw her own father, then the property of being an individual such that he/she/it saw his/her/its own father is an element of the set of properties shared by every girl.

We should also mention that Sportiche (2005) as well as Barker and Shan (2008) differ from Reinhart regarding the role of c-command in the bound-variable readings of pronouns. See (30) below.

To summarize, although the interaction of quantifiers and pronouns raises many important questions and supplies an important set of data, by setting them aside in this book we do not deprive quantifiers of one of their fundamental roles. For general discussion see Büring (2005); Szabolcsi (2008).

2.3.4 Quantifier Raising: May

May’s (1977, 1985) generative syntactic treatment of quantifier scope is the most similar to Montague’s. May first derives a syntactic structure leading to the surface string with quantifier phrases in argument positions. This structure is input to further syntactic rules whose output (Logical Form) feeds only semantic interpretation, not pronunciation. Such a rule is Quantifier Raising (QR), which adjoins quantifier phrases to VP or to S (Tense Phrase, TP in more recent terminology). The scope of the adjoined quantifier phrase is its c-command domain. We simply assume that a phrase c-commands its sister relative to the first branching node above it. Crucial is the consequence that the higher a quantifier is adjoined, the wider scope it takes.

Notice that (27) is parallel to Montague’s (19) and (28) to Montague’s (21). A syntactic difference is that Montague intersperses the steps that disambiguate scope with those that create the surface string, and May does not.

(27)
A difference more important to us is that while May treats the phrases *every man* and *more than one dragon* as normal categorematic expressions in deriving the surface syntax, in his Logical Form these phrases behave like the syncategorematic operators of the predicate calculus: they are co-indexed with the traces left by QR directly, without the mediation of $\lambda$-abstraction. This difference can be eliminated by imagining that there is an abstraction-step hidden between assembling an S and adjoining a quantifier phrase to it, such as $\lambda t_{i}[\text{spot}(t_{i})](t_{k})$ preceding the adjunction of *every man* to S in (27). With that, the parallelism between the two pairs of derivations is essentially complete. Reversing historical order we might look at Montague’s grammar as one that builds the output of May’s compositionally, without invoking movement. Heim and Kratzer (1998) show that a compositional strategy may even include movement. Specifically within the copy theory of movement Fox (2002a,b) reinterprets the lowest copy of QR as a definite description with a bindable variable; see §4.2.2.

Looking back, both May’s and Montague’s scope assignment strategies conform to our basic assumption about scope in (10). May’s is intended to conform to (11), the hypothesis about scope and syntactic domain as well because, May argues, the Logical Form (LF) representations produced by QR are part of syntax. That means that LF representations obey essentially the same principles that govern the well-formedness of syntactic representations that feed Phonetic Form. In May (1985) this claim is in fact only partially borne out. For example, QR indeed obeys certain principles that overt movement does, but it also operates more locally than its semantically plausible overt relative, *wh*-movement. A universal quantifier does not scope out of its tensed clause, but a *wh*-phrase may move out of it tensed clause. With the advent of feature-driven movement in Minimalist syntax (Chomsky 1995) the fact that QR is a non-feature-driven adjunction operation became another point of difference. Such reasons led Hornstein (1995) to propose that QR does not exist, and scope is a by-product of movement motivated by Case-assignment, which is indeed clause-bounded (i.e. operates within the confines of one tensed
Generalized quantifiers and their elements: operators and their scopes

Kennedy (1997) demonstrates that antecedent-contained verb-phrase deletion can only avoid infinite regress if it involves an instance of QR that cannot be motivated by Case; this is a strong argument against Hornstein’s proposal. Are we then back to square one with respect to the clause-boundedness of QR? Cecchetto (2004) points out that Chomsky’s (2001) Phase Theory offers a natural way to accommodate it. A phase is a chunk of structure at the edge of which syntactic memory is emptied, similarly to Cooper’s (1983) theory of quantification, where the so-called quantifier store is obligatorily emptied at the clause-boundary. As of date QR remains part of Minimalist syntax. (See Szabolcsi 2000 and Fox 2002a for a more detailed overview of the syntax of scope, and Chapter 11 for some recent developments.)

It is not too difficult to argue that quantifier scope is syntactically constrained if in doing so one is allowed to postulate syntactic structures and syntactic constraints whose justification comes exclusively, or largely, from matters of interpretation. Among others, such is the constraint that the semantic scope of a linguistic operator coincides with its c-command domain, the most popular version of (11). It turns out that if one carefully discounts all cases where the non-c-commanding operator is also separated from the intended dependent element by a tensed-clause boundary, then evidence for the c-command restriction is slim. Consider for example (29), an example that would demonstrate that it is not sufficient for a quantifier to precede a pronoun to bind it.

(29) That every boy was hungry surprised his mother.

# ‘for every boy, that he was hungry surprised his own mother’

In (29) every boy does not c-command his mother and, indeed, the latter has no bound-variable interpretation. But every boy is also separated from his mother by a tensed-clause boundary, so the latter does not even fall within the quantifier’s scope. It is probably for such reasons that Sportiche (2005) does not make structural c-command a condition on scope and on pronoun binding, contra Reinhart (1979, 1983) and much literature following her; Barker and Shan (2008) advocate a similar revision.

(30) Sportiche’s Principles of Scope:
(i) If X superficially c-commands Y, Y can be interpreted in the scope of X.
(ii) X and Y can outscope each other only if X and Y are clause-mates.

Principle of Pronominal Binding:
A pronoun can behave as a variable bound by X only if it can be interpreted in the scope of X.
2.3 Scope and constituent structure

2.3.5 All the scopes, but a simple syntax: Hendriks

What emerges from the above is that any representation of the $S > O$ and the $O > S$ readings will have to boil down to the schemas in (31)–(32). $P(x)(y)$ is forced by the assumption that the natural language predicates at hand take individuals as arguments. The $\lambda$-binding (predicate-abstraction) steps are forced by the assumption that quantifier phrases denote generalized quantifiers. The two schemas differ as to which argument slot is $\lambda$-bound first and which second.

(31) \[ QP_A(\lambda y[QP_B(\lambda x[P(x)(y)])]) \]  \hspace{1cm} S > O

(32) \[ QP_A(\lambda x[QP_B(\lambda y[P(x)(y)])]) \]  \hspace{1cm} O > S

One of the key insights in Hendriks (1993) is that it is possible to abstract these interpretive schemas away from the specific quantifier phrases $QP_A$ and $QP_B$. This in turn allows one to dissociate the interpretive schema from the syntactic constituent structure of the sentence.

Replace $QP_A$ and $QP_B$ with variables $A$ and $B$ of the same type as generalized quantifiers, $(e, t, t)$, and abstract over them with $\lambda$-operators. Because the variables $A$, $B$ are not individual variables but are of the generalized quantifier type, the $\lambda$-expressions in (33)–(34) take quantifier phrases as arguments, rather than the other way around. The order in which the $\lambda A$ and $\lambda B$ prefixes appear determines the order in which the verb picks up its arguments, but it does not affect their scope, so it can be dictated by independent syntactic considerations; for example we may assume an invariant (S (V O)) structure. In both (33) and (34) the first quantifier phrase the $\lambda$-expression applies to will be the direct object. The relative scope of the quantifier phrases replacing $A$ and $B$ is determined by their relative order within the underlined portions of (33)–(34):

(33) \[ \lambda B\lambda A[A(\lambda y[B(\lambda x[P(x)(y)])])] \]  \hspace{1cm} schema of $S > O$

(34) \[ \lambda B\lambda A[\lambda x[A(\lambda y[P(x)(y)])]] \]  \hspace{1cm} schema of $O > S$

This is a nimbler logic than the first-order predicate calculus; it allows one to arrest the action of a quantifier at the point it enters the formula and to release it where desired. The quantifier’s action is released where it actually applies to an expression that denotes a property. Notice that (33) and (34) fully conform to (10), although they abandon (11).\(^7\)

Where are the schemas in (33)–(34) coming from, if they do not simply record the phrase-by-phrase assembly of the material of the sentence? Hendriks proposes to assign flexible types to verbs, so that two versions of spot for example anticipate two different scope relations between the subject and the object. (33) and (34) are two interpretations for the same transitive verb $P$. Below is a constituent-by-constituent derivation of the
O > S reading. The verb combines with both the direct object and the subject by functional application:

\[
\begin{align*}
\text{Spot}: & \lambda B \lambda \lambda z [B(\lambda z[A(\lambda v[\text{spot'}(z)(v)])])] \\
\text{every man'':} & \lambda Q \forall y[\text{man'}(y) \rightarrow Q(y)] \\
\text{spotted every man':} & \lambda B \lambda \lambda z [B(\lambda z[A(\lambda v[\text{spot'}(z)(v)])])](\lambda Q \forall y[\text{man'}(y) \rightarrow Q(y)]) = \\
& \lambda A[\forall y[\text{man'}(y) \rightarrow A(\lambda v[\text{spot'}(y)(v)])]] \\
\text{more than one dragon':} & \lambda P \exists_{>1} z[\text{dragon'}(z) \land P(z)] \\
\text{more than one dragon spotted every man':} & \lambda A[\forall y[\text{man'}(y) \rightarrow A(\lambda v[\text{spot'}(y)(v)])]] \\
& (\lambda P \exists_{>1} z[\text{dragon'}(z) \land P(z)]) = \\
& \forall y[\text{man'}(y) \rightarrow \exists_{>1} z[\text{dragon'}(z) \land \text{spot'}(y)(z)]]
\end{align*}
\]

This is the gist of Hendriks’s proposal. More generally, he shows two important things. First, the different interpretations for the verb can be obtained systematically by so-called type-change rules, in this case by two applications of Argument Raising, see (36). (33) and (34) are due to two different orders in which the subject and the object slots are raised, cf. the underlined segments. Second, all the logically possible scope relations in an arbitrarily multi-clausal sentence, including extensional–intensional ambiguities, can be anticipated by the use of three type-change rules: Argument Raising, Value Raising, and Argument Lowering. We ignore the last one, which turns an intensional relation into an extensional one between individuals, much like Montague’s meaning postulate pertaining to seek’, because in this book we remain agnostic about the proper treatment of de re/de dicto ambiguities. Below are extensionalized Argument Raising and Value Raising. The simplified version of Value Raising is nothing else than the good old type-raising rule that turns proper names into generalized quantifiers.8

\[
\begin{align*}
(36) & \text{Argument Raising:} \\
& \text{If } \alpha' \text{ is the translation of } \alpha, \text{ and } \alpha'' \text{ is of type } \langle A, \langle b, \langle C, d \rangle \rangle \rangle, \text{ then} \\
& \lambda x_A \lambda w_{\langle(b,d),d \rangle} \lambda y_C[w(\lambda z_b[\alpha''(x)(z)(y)])], \\
& \text{which is of type } \langle A, \langle \langle b, d \rangle, d \rangle, \langle C, d \rangle \rangle, \text{ is also a translation of } \alpha, \text{ where } A \text{ and } C \text{ stand for possibly empty sequences of types such that if } g \text{ is a type, } \langle A, g \rangle \text{ and } \langle C, g \rangle \text{ represent the types } \\
& \langle a_1, \langle \ldots(a_n, g) \ldots \rangle \rangle \text{ and } \langle c_1, \langle \ldots(c_n, g) \ldots \rangle \rangle.
\end{align*}
\]

\text{Simplified by taking } A \text{ and } C \text{ to be empty:} \\
\text{If } \alpha' \text{ is the translation of } \alpha, \text{ and } \alpha'' \text{ is of type } \langle b, d \rangle, \text{ then} \\
\lambda w_{\langle(b,d),d \rangle}[w(\lambda z_b[\alpha''(z)])], \\
\text{which is of type } \langle \langle b, d \rangle, d \rangle, d \rangle, \text{ is also a translation of } \alpha.
2.3 Scope and constituent structure

(37) Value Raising:
If $\alpha'$ is the translation of $\alpha$, and $\alpha'$ is of type $\langle A, b \rangle$, then
$\lambda x A \lambda u_{\langle b, d \rangle}[u(\alpha'(x))]$, which is of type $\langle A, \langle\langle b, d \rangle, d \rangle \rangle$, is also a translation of $\alpha$, where $A$ stands for a possibly empty sequence of types such that if $g$ is a type, $\langle A, g \rangle$ represents the types $\langle a_1, \langle \ldots \langle a_n, g \rangle \ldots \rangle \rangle$.

Simplified by taking $A$ to be empty:
If $\alpha'$ is the translation of $\alpha$, and $\alpha'$ is of type $b$, then $\lambda u_{\langle b, d \rangle}[u(\alpha')]$, which is of type $\langle\langle b, d \rangle, d \rangle$, is also a translation of $\alpha$.

Let us mention two other scope phenomena that involve the dissociation of the chronological order of introducing operators into the syntactic structure from the scope they take, and have been handled in the linguistic literature using pieces of logical machinery that are essentially identical to Hendriks's Argument Raising and Value Raising.

Cresti (1995) analyzes “scope reconstruction” using a combination of generalized-quantifier-type and individual-type variables, to an effect very much like that of Argument Raising. Following Higginbotham (1993), Cresti (1995) splits how many people into two quantifiers. “Reconstruction” is so called because in (i) n-many people is “put back” into a lower position for interpretation.

(38) How many people do you think I should talk to?
(i) ‘for what number $n$, you think it should be the case that there are $n$-many people that I talk to’
(narrow scope, amount reading of how many people)
(ii) ‘for what number $n$, there are $n$-many people $x$ such that you think I should talk to $x$’
(wide scope, individual reading of how many people)

Cresti derives the two readings without actual reconstruction. In the derivations below, $x$ is a trace of type $e$ (individuals), and $X$ is a trace of the same type as n-many people (intensionalized generalized quantifiers).

The latter, higher order variable plays the exact same role here as the variables $A$ and $B$ do in (33) and (34). Working bottom-up, each trace is bound by a $\lambda$-operator to allow the next trace or the moved phrase itself to enter the chain. The lowest position of the chain is always occupied by a trace $x$ of the individual type, but intermediate traces (underlined) may make one switch to the higher type $X$. The scope difference with respect to the intensional operator should is due to the fact that in (39) the switch from $x$ to $X$ takes place within the scope of should, whereas in (40) should has no $X$ in its scope. Note that the direction of functional
application is type-driven. In $X \lambda x[\ldots]$ the first expression applies to the second, whereas in $X \lambda X[\ldots]$ the second applies to the first.

(39) narrow scope:

$$[CP \text{ how many people } \lambda X [IP \ldots \text{think } [CP X \lambda X [IP \ldots \text{should } [VP X \lambda x [VP \ldots x \ldots ]]])]]$$

(40) wide scope:

$$[CP \text{ how many people } \lambda X [IP X \lambda x [IP \ldots \text{think } [CP X \lambda x [IP \ldots \text{should } [VP \ldots x \ldots ]]])]]$$

Moltmann and Szabolcsi (1994) use an idea very much like Value Raising (37) to account for the surprising ‘librarians vary with students’ reading of (41):

(41) Some librarian or other found out which book every student needed.

‘for every student $x$, there is some librarian or other who found out which book $x$ needed’

Every student in the complement can make the matrix subject referentially dependent; but under normal circumstances every NP is known not to scope out of its own clause. Moltmann and Szabolcsi argue that there is no need to assume that here, either. Instead, the clausal complement of $\text{found out}$, i.e. $\text{which book every student needed}$, receives a pair-list reading: ‘for every student, which book did he need’. This “pair-list quantifier” as a whole scopes over the subject of $\text{found out}$, its clause-mate. The result is logically equivalent to scoping every student out on its own. (More on reconstruction in §3.2, and on pair-list readings in §4.1.4.)

Generally, let a “layered” quantifier be a QP that contains one or more other quantifier phrases. Besides pair-list readings, possessive constructions are a good example:

(42) a. every boy’s mother

b. an inhabitant of every city

Let QP$_b$ take wide scope within QP$_a$. Quantifying QP$_a$ into a syntactic domain is logically equivalent to assigning QP$_b$ wide scope over that domain. This is the basis for May’s (1985) treatment of (42) without adjoining the universals to S: he adjoins the wide-scoping QP$_b$ just to QP$_a$. We see that the equivalence is also an empirically welcome result when QP$_a$ is a $\text{wh}$-complement. But reliance on it overgenerates when QP$_a$ is a $\text{that}$-complement.

While neither Cresti nor Moltmann and Szabolcsi use flexible types for verbs, the proposals illustrate the naturalness of the logical tools, Argument Raising and Value Raising, that Hendriks employs. Bittner’s
2.3 Scope and constituent structure

(1993) cross-linguistic semantics systematically exploits similar insights. Bittner’s system differs from Hendriks’s in that it is not designed to make everything possible. Its intention is to distinguish between universal, unmarked interpretations and language-specifically available marked ones.

Inspired by computer science, Barker and Shan (2006) associate linguistic expressions with their possible continuations. A continuation is the skeleton of a syntactico-semantic structure that the expression anticipates participating in. Continuized types are similar to Hendriks’s raised types and to context change potentials in dynamic semantics.

2.3.6 Continuations and scope: Barker and Shan

Barker and Shan’s system rests on this main idea:

(43) The meaning of an expression is the set of its possible “continuations”.

This builds on a rich tradition: Montague’s generalized quantifiers as denotations for noun phrases; Cooper’s (1983) quantifier storage; Hendriks’s verb meanings that anticipate how their dependents will arrange themselves in a particular scopal configuration; dynamic semantics for pronominal anaphora; continuation-passing style in functional programming (Plotkin 1975), etc. Related notions are the sets of alternatives in Hamblin (1973) and Rooth (1992).

Beyond the basic insight, the implementation and its utility depend on what kinds of continuations are catered to. One way of looking at generalized quantifiers is that by virtue of being functions from properties to truth values they anticipate the kind of semantic objects they are going to combine with. Hendriks’s scope grammar additionally makes a head, e.g. a verb anticipate (by Argument Raising), the whole derivation in which its dependents will be arranged in a particular scopal configuration; moreover, the head may anticipate (by Value Raising) its maximal projection being a complement of a higher head. In dynamic semantics, most intuitively sentences anticipate being continued with another sentence and to provide antecedents for pronouns in that sentence. So “anticipation” may pertain to argument structure, to scope, to anaphora, and other things.

Barker and Shan have developed a system whose formalism primarily specializes in scope taking and in the binding of pronouns by quantifiers (i.e. quantificational binding). Given the self-imposed thematic limitations of this volume we ignore binding, but we note that the way binding interacts with scope is essential in assessing the merits or demerits of this system. Regarding scope, one of the important ideas can be expressed in generative syntactic terms as follows:\textsuperscript{9}
Pied piping everywhere
When a phrase XP contains a wide-scoping operator of some sort, it becomes an operator-XP. I.e. it inherits the “operator feature” and behaves accordingly.

To take the simplest cases of pied piping, about which is a PP, but it is also a wh-expression, so it behaves like wh-phrases do in the given language. In English, it will be fronted under the appropriate circumstances, e.g. (those secrets,) about which I cannot speak. It is appropriate to call it a wh-PP. About everyone is a PP, but it is also a quantifier phrase, so it takes scope in the way quantifier phrases do; it is a quantificational PP. About himself is a PP, but it is also an anaphor, so it must find an appropriate binder; it is an anaphor-PP. Generalizing, speak about which is a wh-VP, cf. (those secrets,) speaking about which would be dangerous, see everyone is a quantificational VP, see himself is an anaphor-VP, and so on.10

What does this imply for the way sentences are built step by step? Focusing just on quantifiers, it implies that not only plain everyone must be introduced by some kind of a rule of quantification, but about everyone and see everyone must too. If we take a sentence where all the noun phrases happen to be quantifiers, then all the steps of building the sentence are quantifying-in steps.

Another implication is that the grammar must ensure that the expectation to be continued is passed from smaller expressions to the larger ones they build. A highly simplified picture of Groenendijk and Stokhof’s (1990, 1991) dynamic semantics may be the best linguistic illustration of the “continuation-passing” idea (although as a matter of personal history Barker and Shan were inspired by computer science). One of the descriptive questions dynamic theories seek to answer is how singular indefinites support pronominal anaphora even in the absence of c-command – something that universals do not do:

(45) a. A dragon lumbered to the meadow. It hissed.
    b. Every dragon lumbered to the meadow. # It hissed.

If a dragon is an existentially quantified expression, as we have been assuming with Montague, then it is mysterious how it is linked to the pronoun it despite the latter being outside the quantifier’s scope. Heim (1982) proposes to equate the meaning of an expression with its context-change potential, especially its potential to serve as an antecedent for anaphoric expressions. She also proposes however that a dragon’ is not a quantifier, but an open proposition (one with a free variable). Groenendijk and Stokhof adopt the first, fundamental innovation, but explore the possibility to maintain that indefinites denote quantifiers. For present purposes
the following simplification of the theory will suffice. First, Groenendijk and Stokhof assume that all expressions are associated with the set of their possible continuations. I.e. all sentences will be interpreted in the following format: $\lambda p[\ldots p\ldots]$, where $p$ is a variable over possible continuations. The contributions of indefinites and universals differ as to where they force the continuation variable to be located. To model the fact that indefinites are capable of extending their binding scope over the incoming discourse, the continuation variable will find itself within the indefinite’s scope. To model the fact that universals cannot do the same, the continuation variable will find itself outside the universal’s scope. If pronouns are free variables, this machinery must be supplemented with abstraction over assignments, so the pronoun can be brought within the scope of the indefinite; binding itself is effected by an assignment-switcher. Because this aspect is immaterial to present concerns, I will simply omit these ingredients, although the omission makes the following logically plainly incorrect.\textsuperscript{11} (46)–(47) are the interpretations of the two sentences in (45a) as sets of possible continuations:\textsuperscript{12}

$\lambda p\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land p]$

$\lambda q[\text{hiss}'(x) \land q]$

Sets of continuations are combined by functional composition (whether the two clauses are joined by and or simply form a sequence). Functional composition has two important consequences. First, the core of (47) fills an argument slot in (46), that of the conjunct $p$. Second, (47) passes its own expectation to be continued on to the result.

$\lambda p\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land p] \circ \lambda q[\text{hiss}'(x) \land q] =$

$\lambda r[\lambda p[\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land p]][\lambda q[\text{hiss}'(x) \land q](r)]] =$

$\lambda r\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land \text{hiss}'(x) \land r]$

What if we want to finish our story? To eliminate the possibility of further continuations and to obtain a traditional sentence denotation (true or false), the context-change potential is applied to (as opposed to getting composed with) $\text{Truth}$, the tautologous continuation. This has the desired effect, because for any $p$, $p \land \text{Truth} = p$.

$\lambda r\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land \text{hiss}'(x) \land r](\text{Truth}) =$

$\exists x[\text{dragon}'(x) \land \text{lumber}'(x) \land \text{hiss}'(x)]$

Returning to Barker and Shan, in addition to treating all expressions that contain quantifiers as quantificational, they treat the quantificational aspects of expressions in the continuation-passing style just illustrated. To close off the continuation-scope of an expression they apply it to an identity function. This plays the same role as $\text{Truth}$ above: when an $n$-ary
function $f$ is applied to the identity function, the first argument of $f$ is eliminated.

The remarks above, together with the overview of Hendriks’s technique in §2.3.5, provide sufficient background for a brief illustration of how the general ideas are implemented in Barker and Shan (2008) using what the authors call the “tower notation”. This is essentially a simple and transparent proof-theoretic tool that enables one to calculate the results of combination quickly and efficiently. The “tower-operations” can be equivalently spelled out using $\lambda$-expressions.

All expressions are represented as towers with at least three levels: syntactic category, expression, logical form. The lexical items *ran* and *someone* will start out as below. (The system handles quantifier phrases like *some dragon*; we use *someone* just to simplify the exposition.)

The two layers of the downstairs level of *someone* specify the argumental role and the scopal, or continuational, contribution of *someone*. “Below the dashed line” *someone* contributes a variable of type $e$ (here: $y$). “Above the dashed line” it scopes over what will end up enclosed by the brackets $[ ]$, and its contribution as a quantifier is to bind $y$ by $\exists y$. The two layers of the upstairs level match these item-for-item. The category of *someone* is DP. *Someone* will scope over an S (the righthand-side one) and yield an S as a result (the lefthand-side one). That is, (51) conveys the following information about the behavior of *someone*:

\[
\begin{array}{c}
\text{DP\backslash S} \\
\text{\quad ran} \\
\text{\quad ran} \\
\end{array}
\]

\[
\begin{array}{c}
\text{\quad S \downarrow S} \\
\text{\downarrow \quad DP} \\
\text{\quad someone} \\
\text{\quad \exists y[,]} \\
\text{\quad \quad y} \\
\end{array}
\]
The factoring of the contribution of someone into an argumental and a scopal (continuational) part is inspired by Cooper (1983). Cooper was the first to introduce the idea that each quantifier fills an argument place in syntax, but its quantificational content is stored away. Both upstairs and downstairs the “above the dashed line” layers correspond to storage. Barker and Shan’s system is more flexible than Cooper’s in that the meanings of smaller units, e.g. quantificational determiners, can also be defined as storable.

The grammar has two type-shifters: Lift and Lower, and two ways of combination: Scope/ and Scope\ . Lift, Scope, and Lower may apply to the whole item or just to its ground-floor level, with crucially different scope effects. The working of the rules is defined using the tower notation, but it is also spelled out using λ-terms.

The λ-expressions make it clear that Lift is nothing but the Montague-rule, or Hendriks’s Value Raising. Lower applies its input function \( F \) to the identity function, as explained above. \( Exp \) abbreviates expression.

(53) Lift: \( \lambda x \lambda k[k(x)] \)

\[
\begin{array}{c}
A \\
\hline
\text{Exp} \\
x
\end{array}
\Rightarrow
\begin{array}{c}
\hline
\text{Exp} \\
\hline
\text{[]} \\
x
\end{array}
\begin{array}{c}
B \\
\hline
\text{A} \\
\hline
\text{[]} \\
x
\end{array}
\]

(54) Lower: \( \lambda F[F(\lambda x[x])] \)

\[
\begin{array}{c}
\hline
\text{Exp} \\
\hline
f[\text{[]}] \\
x
\end{array}
\Rightarrow
\begin{array}{c}
\hline
\text{Exp} \\
\hline
\text{f[x]}
\end{array}
\begin{array}{c}
\hline
\text{S} \\
\hline
\text{S} \\
\text{Exp} \\
\hline
\text{x}
\end{array}
\begin{array}{c}
\hline
\text{A} \\
\hline
\text{Exp} \\
\hline
\text{x}
\end{array}
\]

Lift maps one-layered levels to two-layered levels. At the upstairs level of the tower Lift takes an expression of category \( A \) and turns it into something that scopes over a \( B \) and yields a \( B \), cf. the well-known lifted categories \( (B/A)\backslash B \) and \( B/(A\backslash B) \). At the downstairs level Lift takes the basic interpretation of \( exp \), \( x \), which matches its basic category \( A \), and yields something that contributes the variable \( x \) but scopes over \( [\text{[]} \) . In contrast, Lower collapses two-layered levels into one-layered levels (in Cooper’s terms, retrieves quantifier meanings from storage).
The two Scope rules only differ from each other syntactically: Scope/ puts together a rightward-looking function of category $B/A$ with its argument of category $A$, whereas Scope\ puts together a leftward-looking function of category $A\setminus B$ with its argument of category $A$. Recall now that Barker and Shan treat all expressions as sets of continuations, and that they combine any two expressions qua quantifiers. As the $\lambda$-expressions in (55) below show, the core of each Scope rule is $k(fx)$, where function $f$ applies to argument $x$, and a higher function $k$ applies to the result. $f$ and $x$ correspond to the two expressions combined. $k$ corresponds to the anticipated continuation. Because Scope takes its input expressions to be quantifiers, it does not combine any $f$ and $x$ directly. Instead, in Scope/ a Left-expression $L$ is quantified into $\lambda f[R(\lambda x[k(fx)])]$, and a right-expression $R$ is quantified into $\lambda x[k(fx)]$. This implies that if the Left-expression and the Right-expression are not originally quantifiers (are not yet expecting to be continued), they must get lifted before they get combined.

The reader who has worked through §2.3.5 will immediately notice the tell-tale signs of two instances of Argument Raising and one instance of Value Raising in the $\lambda$-expression that explicates Scope (see also the endnote with the derivations for some student borrowed every book'). Speaking in Hendriksese, the presence of $k$ comes from Value Raising, and the quantifications involving $L$ and $R$ indicate that Argument Raising has been applied twice in the definition of Scope. It should be born in mind, though, that Barker and Shan do not use Argument Raising and Value Raising as rules in their grammar; the similarities with Hendriks consist in what general logical operations are needed to achieve a particular kind of result. Also, whereas in Hendriks’s grammar AR and VR operate on linguistic expressions like spot, in (55) they operate on the basic combinator that applies $f$ to $x$ and derive a more complex combinator.

In the towers the “below-the-dashed-line” layers of both the downstairs and the upstairs levels continue to pertain to argument structure. In Scope/, $f$ of category $B/A$ applies to $x$ of category $A$ to yield $f(x)$ of category $B$. In the “above-the-dashed-line” layers the Right-expression (here the argument) is placed within the scope of the Left-expression (the function). Notice that $g$ is whatever semantic content $f$ affixes to its scope, and $h$ is whatever semantic content $x$ does. (In (55) the distinct category labels $C$, $D$, and $E$ serve to make transparent that $E$ is the label of the scope of $x$, $D$ the label of the scope of $f$, and $C$ the label of the result of putting the two together.)

(55) a. Scope/ $\lambda L\lambda R\lambda k[L(\lambda f[R(\lambda x[k(f(x)])])])$
2.3 Scope and constituent structure

What we have already suffices to derive Someone ran, as well as a two-quantifier example on its direct, left-to-right scopal reading. We start by lifting ran and combine it with someone by Scope\.

\((56)\) \(\lambda x \lambda k[x](\text{ran}) = \lambda k[k(\text{ran})]\)

\((57)\) \(\lambda L \lambda R \lambda k[L(\lambda x[R(\lambda f[k(f(x))])])(\lambda P \exists y[P y])(\lambda k[k(\text{ran})]) = \lambda k \exists y[k(\text{ran}(y))]\)
The result is a set of continuations. To complete the derivation and obtain a traditional sentence, Lower applies. It collapses the “above the line” and the “below the line” material by inserting the latter into the former.

\[ (59) \lambda F[(\lambda x. x)((\lambda k \exists y (k(ran(y)))) = \\
\lambda k \exists y (k(ran(y)))(\lambda x. x) = \\
\exists y (ran(y)) \]

\[
\begin{array}{c}
\text{S} \\
\text{S} \\
\text{someone ran} \\
\exists y, [ ] \\
\text{ran(y)} \\
\Rightarrow \\
\text{S} \\
\text{someone ran} \\
\exists y, [ran(y)]
\end{array}
\]

The derivation of Someone loves everyone on its direct scopal reading involves the same steps. First loves is Lifted. Then lifted loves combines with everyone by Scope/. The result combines with someone by Scope\/. Finally, Lower eliminates the possibility of further continuations. Below is a compressed derivation:

\[ (59) \text{Direct scope (Barker and Shan 2008)} \]

\[
\begin{array}{c}
\text{S} \downarrow \text{S} \downarrow \text{DP} \\
\text{someone} \\
\exists x, [ ] \\
x \\
\Rightarrow \\
\text{S} \downarrow \text{S} \downarrow (\text{DP} \backslash \text{S}) / \text{DP} \\
\text{loves} \\
[ ] \\
\forall y, [ ] \\
y \\
\text{S} \downarrow \text{S} \downarrow \text{DP} \\
\text{everyone} \\
\forall y, [ ] \\
y = \\
\text{S} \downarrow \text{S} \downarrow \text{S} \downarrow \text{S} \downarrow \text{DP} \\
\text{someo. loves everyo.} \\
\exists x. \forall y, [ ] \\
\forall y, \text{loves(y)}(x) \\
\Rightarrow \\
\text{S} \downarrow \text{S} \downarrow \text{S} \downarrow \text{S} \downarrow \text{S} \downarrow \text{DP} \\
\text{someo. loves everyo.} \\
\exists x. \forall y, \text{loves(y)}(x)
\end{array}
\]

In (59) someone acquires wide scope over everyone because Scope/ inserts \( \forall y, [ ] \) into the bracketed space representing the scope of \( \exists x, [ ] \). In this grammar inverse scope cannot be obtained by performing the same steps in the reverse order, which is what happens in the scope grammars reviewed above. Instead, both the inverse-scope-taker and the inversely-
...and so on. Many of these topics are included in Barker and Shan arguing out of a quantifier phrase, the parasitic scope of the adjective pronominal binding (Superiority, Cross-over), donkey-anaphora and Shan argue, that shows the empirical significance of the very thing

If the default evaluation order tracks left-to-right order, as Barker structural condition like c-command is associated with precedence in evaluation. The result will be lowered twice to complete the derivation.\textsuperscript{14}

(60) Inverse scope (Barker and Shan 2008)

\begin{align*}
\text{S} & \quad \text{S} \\
\text{S} & \quad \text{S} \\
\text{DP} & \\
\text{} & \quad [ ] \\
\text{} & \quad \exists x [ ] \\
\text{} & \quad x \\
\text{someone} & \\
\text{loves} & \\
\text{} & \quad y \\
\text{everyone} & \\
\text{loves} & \\
\text{} & \quad [ ] \\
\text{∀y} [ ] \\
\end{align*}

Barker and Shan have applied these basic ideas, in joint and/or separate publications, to the interaction of quantifiers and \textit{wh}-phrases with pronominal binding (Superiority, Cross-over), donkey-anaphora and binding out of a quantifier phrase, the parasitic scope of the adjective \textit{same}, and so on. Many of these topics are included in Barker and Shan (2009).

Moortgat’s (1996) \textit{q} type-constructor and the continuation semantics for symmetric categorial grammar in Bernardi and Moortgat (2007) represent convergent ideas; see Bernardi (2010) for discussion designed to be friendly to the linguist reader. In both these theories scope and binding are dependent on the order in which expressions are evaluated. This can be read off the way in which the sentences are constructed, but no structural condition like c-command is associated with precedence in evaluation. If the default evaluation order tracks left-to-right order, as Barker and Shan argue, that shows the empirical significance of the very thing

\begin{align*}
\text{S} & \quad \text{S} \\
\text{S} & \quad \text{S} \\
\text{} & \quad \forall y [ ] \\
\text{} & \quad \exists x [ ] \\
\text{someo. loves everyo.} & \\
\text{loves(y)(x)} & \\
\text{∀y.∃x.loves(y)(x)} & \\
\end{align*}
that the introduction of c-command was meant to minimize, although it
does not bring back left-to-right order as an inviolable condition.

### 2.4 Summary and Direct Compositionality

Section §2.2 presented DP-denotations as generalized quantifiers: sets of
properties (extensionally, sets of sets-of-individuals). The scope of a quan-
tifier $A$ is the property that is asserted to be an element of $A$ on a given
derivation of the sentence. If that property incorporates another operator
$B$ (quantifier, negation, modal, etc.), then $A$ automatically takes scope
over $B$. The general lesson of Section §2.3 is that there are many differ-
et ways to implement this scenario. It may be acted out in the syntactic
derivation of the sentence, but it may as well be squeezed into the flex-
tible types of the participating expressions. Consequently, we may create
abstract constituents by movement, but we may alternatively stick to
some independently motivated constituent structure. We may bind syn-
tactic variables (empty categories, traces), as Montague or May, but we
may alternatively do without them and go “variable free”, as Hendriks
or Barker and Shan.\(^{15}\) Notably, both Hendriks’s and Barker and Shan’s
scope grammars are directly compositional, a property advocated in Ja-
cobson (2002). Direct Compositionality means that each constituent built
by the independently motivated syntax is immediately assigned its final
and explicit interpretation.

The fact that one can take either approach is good news. But having to
choose between them may not be so good, since both approaches offer their
own insights. Barker (2007) makes the very important claim that it is in
fact not necessary to choose. Building on Jäger’s (2005) proof-theoretical
proposal Barker points out that a grammar can deliver direct composi-
tionality “on demand”. Here the long-distance (Montague/May/Heim and
Kratzer-style) and the local (Hendriks-style) analyses arise from one and
the same set of rules, none of which are redundant. For every derivation in
which an expression is bound at a distance or takes wide scope, there will
be an equivalent derivation in which the semantic contribution of each con-
stituent is purely local. As Barker explains, the interconvertibility follows
from a natural symmetry in the grammar itself. The symmetry concerns
rules of use and rules of proof in the Gentzen calculus (Gentzen 1935).
Roughly, rules of use connect expressions directly over long distances, and
embody the global view. Rules of proof help characterize the contribution
of individual expressions within a complex constituent. Barker enriches
Jäger’s grammar and introduces rules of disclosure, which establish an ex-
plicit connection between the long-distance semantic effect of an element
and its local denotation.