Scope as Syntactic Abstraction

Chris Barker(✉)

New York University, New York, NY 10003, USA
chris.barker@nyu.edu
http://files.nyu.edu/cb125/public/

Abstract. What is the logic of scope? By "scope", I mean scope-taking in natural languages such as English, as illustrated by the sentence *Ann saw everyone*. In this example, the quantifier denoted by *everyone* takes scope over the rest of the sentence, that is, it takes the denotation of the rest of the sentence as its semantic argument: **everyone**(λx.**saw**(x)(**ann**)). The answer I will give here will be to provide a substructural logic whose two modes are related by a single structural postulate. This postulate can be interpreted as constituting a kind of lambda-abstraction over structures, where the abstracted structures are interpreted as delimited continuations. I discuss soundness and completeness results, as well as cut elimination. I also compare the logic to a number of alternative approaches, including the standard technique of Quantifier Raising, and mention applications to scope ambiguity and parasitic scope.

Keywords: Scope · Continuations · Substructural logic · Quantifier raising · Parasitic scope · Natural language quantification

1 What is the Logic of Scope?

Just as we might ask "What is the logic of negation?", we might ask "What is the logic of scope?". And just as the first question has many answers, so too will the second. The answer I will give here will take the form of a substructural logic containing a single structural postulate. I will suggest this logic characterizes a kind of scope-taking that has applications in the analysis of natural language.

1.1 Scope in Natural Language

Many natural languages have scope-taking expressions, including English:

(1) Ann saw everyone.

In (1), the denotation of the quantifier *everyone* takes the rest of the sentence in which it occurs as its semantic argument. That is, the denotation of the sentence as a whole is given by **everyone**(λx.**saw**(x)(**ann**)).

There are three important properties of scope-taking in natural language that I will discuss here: unbounded scope displacement, embedded scope-taking, and scope ambiguity (see [4] for a more complete discussion).

T. Murata et al. (Eds.): JSAI-isAI 2014 Workshops, LNAI 9067, pp. 184–199, 2015.
DOI: 10.1007/978-3-662-48119-6_14

(2) Ann saw the mother of everyone's lawyer.

In (2), despite being embedded inside of two layers of possessive constructions, the quantifier still takes scope over the entire sentence. In general, there is no upper limit to the structural distance over which an expression can take scope.

(3) a. Bill thinks [Ann saw everyone].
b. $\mathbf{thinks}(\forall x.\mathbf{saw}(x)(\mathbf{ann}))(\mathbf{bill})$

However, in (3a), the quantifier takes scope only over the [bracketed] embedded clause *Ann saw everyone*, which is a proper subpart of the complete sentence. The fact that scopal elements can take embedded scope is what makes *undelimited* continuations unsuited to modeling scope (see Chap. 18 of [5] for discussion); *delimited* continuations are a better fit.

(4) a. Someone loves everyone.
b. $\exists x \forall y.\mathbf{loves}(y)(x)$
c. $\forall y \exists x.\mathbf{loves}(y)(x)$

Scope ambiguity can arise when there is more than one quantifier in the sentence. There can in general be as many as $n!$ distinct denotations, where n is the number of quantifiers.

1.2 Quantifier Raising

By far the dominant way to think about scope-taking is Quantifier Raising (QR), as discussed in detail in [8]. Quantifier Raising accounts for unbounded scope displacement, embedded scope-taking, and scope ambiguity.

From a logical point of view, Quantifier Raising can be seen as a structural relation. That is, Quantifier Raising reconfigures a logical structure by moving the quantifier to adjoin to its scope domain, placing a variable in the original position of the quantifier, and abstracting over the variable at the level of the scope domain.

$$[\text{Ann } [\text{called everyone}]] \overset{\text{QR}}{\Rightarrow} [\text{everyone}(\lambda x[\text{Ann } [\text{called } x]])]$$

Here, the scope domain of *everyone* is the entire clause.

Because the QR operation can target embedded S nodes, embedded scope falls out naturally. Just as naturally, QR easily accounts for scope ambiguity by allowing QR to target quantifiers in any order.

$$\begin{aligned}
\text{Linear scoping}: \quad & [\text{someone } [\text{called everyone}]] \\
\Rightarrow & [\text{everyone}(\lambda x[\text{someone } [\text{called } x]])] \\
\Rightarrow & [\text{someone}(\lambda y[\text{everyone}(\lambda x[y \ [\text{called } x]])])] \\
\text{Inverse scoping}: \quad & [\text{someone } [\text{called everyone}]] \\
\Rightarrow & [\text{someone}(\lambda y[y \ [\text{called everyone}]])] \\
\Rightarrow & [\text{everyone}(\lambda x[\text{someone}(\lambda y[y \ [\text{called } x]])])]
\end{aligned}$$

Raising the direct object first and then the subject gives linear scope, and raising the subject first and then the direct object gives inverse scope.

So far, so good. What remains to be done is to characterize Quantifier Raising from a logical point of view. This is what the remainder of this paper sets out to do (see especially the discussion in Sect. 6.4).

1.3 The *q* Type Constructor

[14] extends Lambek grammar with a type constructor q ('q' for 'quantification') which takes three categories as parameters and has the following logical behavior:

$$\frac{\Gamma[A] \vdash B \qquad \Sigma[C] \vdash D}{\Sigma[\Gamma[q(A,B,C)]] \vdash D}\, q \tag{5}$$

An expression in category $q(A, B, C)$ functions locally (i.e., with respect to the context $\Gamma[\,]$) as an A, takes scope over a structure in category B, and allows the structure over which it takes scope to function in the larger context (i.e., with respect to $\Sigma[\,]$) as an expression of category C. This is exactly what a scope-taking expression needs to do, and any adequate account of scope in natural language should account for the ground covered by q.

However, from a logical point of view, q is problematic. For instance, although it is easy to write a left rule (a rule of use) for q, as in (5), a general right rule (a rule of proof) remains elusive (see [16]). As I will explain below in Sect. 6.1, the resolution of this puzzle here will be to factor the q inference into the interaction of the structural postulate with two independent logical inferences, each of which has its own left and right rules.

1.4 What this Logic for Scope Will not Account for

The account here seeks only to characterize an idealized, unconstrained version of quantifier scope. In any natural language, scope-taking will be constrained by syntactic and lexical factors. See [6] or [10] for formal grammars (also based on delimited continuations) that propose principled constraints on scope-taking.

2 $\mathbf{NL}_{\boldsymbol{\lambda}}$

The substructural grammar for characterizing scope discussed here is based on the non-associative Lambek grammar NL (see, e.g., [13,17]). Since NL rejects all structural rules, including exchange, there will be two versions of implication: $\backslash$, in which the argument is on the left, and $/$, in which the argument is on the right.

NL characterizes the logic of function/argument combination when the functor is linearly adjacent to the argument. However, for scope-taking, linear adjacency is not sufficient. After all, a scope-taker is not adjacent to its argument—it is contained *within* its argument. What we need is a syntactic notion of 'surrounding' and 'being surrounded by'. Therefore the grammar here will provide two modes: not only a merge mode (already introduced), for ordinary function/argument combination, with implications $\backslash$ and $/$; but also a continuation mode, which will govern scope-taking, with implications $\mathbin{\backslash\!\!\backslash}$ and $\mathbin{/\!\!/}$. (The interpretation of the continuation mode will be explained shortly).

The logical rules for these connectives are identical to the rules given in [17]:129. They constitute the logical core of a two-mode type-logical grammar:

$$\frac{}{A \vdash A}\,\text{Axiom} \tag{6}$$

$$\frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[\Gamma{\cdot}A \backslash B] \vdash C}\backslash L \qquad \frac{A{\cdot}\Gamma \vdash B}{\Gamma \vdash A \backslash B}\backslash R \qquad \frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[B/A{\cdot}\Gamma] \vdash C}/L \qquad \frac{\Gamma{\cdot}A \vdash B}{\Gamma \vdash B/A}/R$$

$$\frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[\Gamma \circ A \mathbin{\backslash\!\backslash} B] \vdash C}\mathbin{\backslash\!\backslash} L \qquad \frac{A \circ \Gamma \vdash B}{\Gamma \vdash A \mathbin{\backslash\!\backslash} B}\mathbin{\backslash\!\backslash} R \qquad \frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[B \mathbin{/\!/} A \circ \Gamma] \vdash C}\mathbin{/\!/} L \qquad \frac{\Gamma \circ A \vdash B}{\Gamma \vdash B \mathbin{/\!/} A}\mathbin{/\!/} R$$

The sequents in the logical rules above have the form $\Gamma \vdash A$, where A is a category and Γ is a structure. All categories are structures, and if Γ and Δ are structures, then $\Gamma{\cdot}\Delta$ (merge mode) and $\Gamma \circ \Delta$ (continuation mode) are also structures.

In order to allow expressions to combine with material that surrounds it (or that it surrounds), we need to add a structural rule. In order to state this structural rule, we will need to enlarge the set of structures to include **gapped structures**: if $\Sigma[\Delta]$ is a structure containing a distinguished substructure Δ, then $\lambda\alpha\, \Sigma[\alpha]$ is also a structure, where α is a variable taken from the set $x, y, z, \ldots$. For instance, $\lambda x\, x$, $\lambda y\, y$, $\lambda x\, (x{\cdot}\text{left})$, $\lambda x\, (\text{John}{\cdot}(\text{saw}{\cdot}x))$, and $\lambda x\, \lambda y\, (y{\cdot}(\text{saw}{\cdot}x))$ are gapped structures.

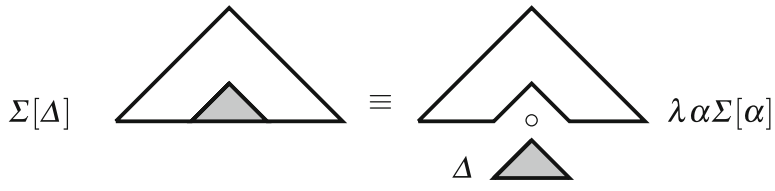
Although gapped structures have important predecessors, including [7,19], they are not standard in discussions of substructural logics. One of the main goals of this paper is to explain how to understand gapped structures. A crucial part of achieving this goal will be to introduce a second substructural logic in the next section, NL_{CL}, which will be equivalent to (a restricted version of) NL_{λ}. NL_{CL} is a standard substructural logic, and does not involve any gapped structures.

With gapped structures in hand, we can state the following structural inference rule:

$$\frac{\Gamma[\Sigma[\Delta]] \vdash A}{\Gamma[\Delta \circ \lambda\alpha\, \Sigma[\alpha]] \vdash A}\,\lambda \tag{7}$$

In words: if a structure Σ contains within it a structure Δ, then Δ can take scope over the rest of Σ, where 'the rest of Σ' is represented as the gapped structure $\lambda\alpha\, \Sigma[\alpha]$.

Schematically, we have:



(8)

The postulate says that if Δ (the small grey triangle) is some structure embedded within a larger structure Σ (the complete larger triangle), we can view these components in a completely equivalent way by articulating them into a foreground and a background, that is, into a plug and a context—an expression and its continuation. Then Δ will be the foregrounded expression, and the clear notched triangle will be its context, the continuation $\lambda\alpha\Sigma[\alpha]$.

An expression in a category with the form $A\backslash\!\backslash B$ is a *continuation*: something that would be a complete expression of category B, except that it is missing an expression of category A somewhere inside of it. An expression in a category with the form $C/\!\!/(A\backslash\!\backslash B)$ will be something that combines with a continuation of category $A\backslash\!\backslash B$ surrounding it to form a result expression of category C.

This logic allows for unbounded scope displacement, since there are no constraints on the complexity of the scope host Σ. It also allows for embedded scope-taking, since Γ may be non-empty. As for scope ambiguity, we have the following two derivations:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{DP}\cdot(\mathrm{loves}\cdot\mathrm{DP})\vdash \mathrm{S}}{\mathrm{DP}\circ\lambda x(\mathrm{DP}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}\lambda}{\lambda x(\mathrm{DP}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{DP}\backslash\!\backslash \mathrm{S}}\backslash\!\backslash R \quad \mathrm{S}\vdash \mathrm{S}}{\mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\backslash \mathrm{S})\circ\lambda x(\mathrm{DP}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}/\!\!/ L}{\mathrm{everyone}\circ\lambda x(\mathrm{DP}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}\mathrm{LEX}}{\mathrm{DP}\cdot(\mathrm{loves}\cdot\mathrm{everyone})\vdash \mathrm{S}}\lambda}{\mathrm{DP}\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{everyone}))\vdash \mathrm{S}}\lambda}{\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{everyone}))\vdash \mathrm{DP}\backslash\!\backslash \mathrm{S}}\backslash\!\backslash R \quad \mathrm{S}\vdash \mathrm{S}}{\mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\backslash \mathrm{S})\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{everyone}))\vdash \mathrm{S}}/\!\!/ L}{\mathrm{someone}\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{everyone}))\vdash \mathrm{S}}\mathrm{LEX}}{\mathrm{someone}\cdot(\mathrm{loves}\cdot\mathrm{everyone})\vdash \mathrm{S}}\lambda$$

The Curry-Howard labeling for this derivation (see [5]) is $\exists x\forall y.\mathbf{loves}\ y\ x$. In general, the scope-taker that is focussed (i.e., targeted by the structural postulate) lower in the proof takes wider scope.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{DP}\cdot(\mathrm{loves}\cdot\mathrm{DP})\vdash \mathrm{S}}{\mathrm{DP}\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{DP}))\vdash \mathrm{S}}\lambda}{\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{DP}))\vdash \mathrm{DP}\backslash\!\backslash \mathrm{S}}\backslash\!\backslash R \quad \mathrm{S}\vdash \mathrm{S}}{\mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\backslash \mathrm{S})\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{DP}))\vdash \mathrm{S}}/\!\!/ L}{\mathrm{someone}\circ\lambda x(x\cdot(\mathrm{loves}\cdot\mathrm{DP}))\vdash \mathrm{S}}\mathrm{LEX}}{\mathrm{someone}\cdot(\mathrm{loves}\cdot\mathrm{DP})\vdash \mathrm{S}}\lambda}{\mathrm{DP}\circ\lambda x(\mathrm{someone}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}\lambda}{\lambda x(\mathrm{someone}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{DP}\backslash\!\backslash \mathrm{S}}\backslash\!\backslash R \quad \mathrm{S}\vdash \mathrm{S}}{\mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\backslash \mathrm{S})\circ\lambda x(\mathrm{someone}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}/\!\!/ L}{\mathrm{everyone}\circ\lambda x(\mathrm{someone}\cdot(\mathrm{loves}\cdot x))\vdash \mathrm{S}}\mathrm{LEX}}{\mathrm{someone}\cdot(\mathrm{loves}\cdot\mathrm{everyone})\vdash \mathrm{S}}\lambda$$

In this case, the semantic labeling gives the universal wide scope: $\forall y\exists x.\mathbf{loves}\ y\ x$.

3 Soundness and Completeness via NL_{CL}

The proofs of soundness and completeness for NL_λ will proceed by defining NL_{CL}, a more standard substructural logic whose soundness and completeness follows from the general results of [20]. I will then give conditions under which NL_λ and NL_{CL} are equivalent.

NL_{CL} has the same logical rules as NL_λ. Instead of the structural postulate λ, however, NL_{CL} has the following three structural postulates:

$$\frac{p}{p \circ \mathsf{I}}\,\mathsf{I} \qquad\qquad \frac{p\cdot(q \circ r)}{q \circ ((\mathsf{B}\cdot p)\cdot r)}\,\mathsf{B} \qquad\qquad \frac{(p \circ q)\cdot r}{p \circ ((\mathsf{C}\cdot q)\cdot r)}\,\mathsf{C} \tag{9}$$

These postulates are identical to the ones given in [2]. [20]:30 considers I (which he writes '0') as "a zero-place punctuation mark," where punctuation marks (p. 19) "stand to structures in the same way that connectives stand to formulae." Likewise, B and C are also zero-place punctuation marks. The double horizontal line indicates that these rules are bi-directional, i.e., inference in the top-to-bottom direction and in the bottom-to-top direction are both valid. Restall calls the top-to-bottom inference for the I postulate Push, and the other direction Pop.

In the form of an official inference rule, the I postulate (for instance) is written

$$\frac{\Sigma[p] \vdash A}{\Sigma[p \circ \mathsf{I}] \vdash A}, \tag{10}$$

and similarly for the other rules.

An example derivation will show how these postulates work together to achieve in-situ quantification for the sentence *John saw everyone*:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{DP}\vdash\mathrm{DP}\quad\dfrac{\mathrm{DP}\vdash\mathrm{DP}\quad\mathrm{S}\vdash\mathrm{S}}{\mathrm{DP}\cdot\mathrm{DP}\backslash\mathrm{S}\vdash\mathrm{S}}\,\backslash L}{\mathrm{DP}\cdot((\mathrm{DP}\backslash\mathrm{S})/\mathrm{DP}\cdot\mathrm{DP})\vdash\mathrm{S}}\,/L}{\text{john}\cdot(\text{saw}\cdot\mathrm{DP})\vdash\mathrm{S}}\,\mathrm{LEX}}{\text{john}\cdot(\text{saw}\cdot(\mathrm{DP}\circ\mathsf{I}))\vdash\mathrm{S}}\,\mathsf{I}}{\text{john}\cdot(\mathrm{DP}\circ((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I}))\vdash\mathrm{S}}\,\mathsf{B}}{\mathrm{DP}\circ((\mathsf{B}\cdot\text{john})\cdot((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I}))\vdash\mathrm{S}}\,\mathsf{B}}{(\mathsf{B}\cdot\text{john})\cdot((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I})\vdash\mathrm{DP}\backslash\!\!\backslash\mathrm{S}}\,\backslash\!\!\backslash R\quad\mathrm{S}\vdash\mathrm{S}}{\mathrm{S}/\!\!/(\mathrm{DP}\backslash\!\!\backslash\mathrm{S})\circ((\mathsf{B}\cdot\text{john})\cdot((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I}))\vdash\mathrm{S}}\,/\!\!/L}{\text{everyone}\circ((\mathsf{B}\cdot\text{john})\cdot((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I}))\vdash\mathrm{S}}\,\mathrm{LEX}}{\text{john}\cdot(\text{everyone}\circ((\mathsf{B}\cdot\text{saw})\cdot\mathsf{I}))\vdash\mathrm{S}}\,\mathsf{B}}{\text{john}\cdot(\text{saw}\cdot(\text{everyone}\circ\mathsf{I}))\vdash\mathrm{S}}\,\mathsf{B}}{\text{john}\cdot(\text{saw}\cdot\text{everyone})\vdash\mathrm{S}}\,\mathsf{I} \tag{11}$$

NL_{CL} is sound and complete with respect to the usual class of relational models. This follows directly from the proofs given in [20], Chap. 11. In particular, [20]:249 provides an algorithm for constructing frame conditions corresponding to the structural postulates.

Theorem (Soundness and Completeness): $X \vdash A$ is provable in NL_{CL} iff for every model $\mathfrak{M} = \langle \mathscr{F}, \models \rangle$ that satisfies the frame conditions, $\forall x \in \mathscr{F}, x \models X \rightarrow x \models A$.

Proof: given in [20], theorems 11.20, 11.37.

Furthermore, NL_{CL} is conservative with respect to NL. That is,

Theorem (Conservativity): Let an NL sequent be a sequent built up only from the formulas and structures allowed in NL: $/, \backslash, \cdot$. An NL sequent is provable in NL_{CL} iff it is provable in NL.

See [5] for details.

4 The Connection Between NL_{λ} and NL_{CL}

This section investigates the conditions under which a derivation in NL_{λ} has an equivalent derivation in NL_{CL}.

I define the following class of structures:

$$\Gamma\lceil p\rceil ::= \quad p \quad | \quad p \circ q \quad | \quad q \cdot \Gamma\lceil p\rceil \quad | \quad \Gamma\lceil p\rceil \cdot q \quad | \quad \lambda y.\, \Gamma\lceil p\rceil \tag{12}$$

Given a structure p, a $\lceil\ \rceil$-context will consist either of the empty context, or else the entire left element at the top level of a $\circ$ structure, or else a larger context built up from $\cdot$ and λ. We can impose these restrictions on NL_{λ} by replacing the original lambda postulate with one that mentions $\lceil\ \rceil$-contexts:

$$\Sigma\lceil\Delta\rceil \equiv \Delta \circ \lambda\alpha\, \Sigma\lceil\alpha\rceil \tag{13}$$

To illustrate, the following (bidirectional) inferences are licensed by (13):

$$\frac{A}{A \circ \lambda x x} \qquad \frac{A \circ B}{A \circ \lambda x (x \circ B)} \qquad \frac{A \cdot B}{A \circ \lambda x (x \cdot B)} \qquad \frac{\lambda x.(x \cdot B)}{B \circ \lambda y \lambda x (x \cdot y)} \tag{14}$$

But not these:

$$\frac{(A \cdot B) \circ C}{A \circ \lambda x ((x \cdot B) \circ C)} \qquad \frac{A \circ B}{B \circ \lambda y (A \circ y)} \tag{15}$$

The reason these last two inferences are not allowed is that abstraction across $\circ$ is forbidden unless the abstractee is the complete left element connected by $\circ$.

The inspiration for NL_{CL} comes from the well-known equivalence between the lambda calculus and Combinatory Logic. More specifically, the postulates of NL_{CL} implement a version of Shönfinkel's embedding of λ-terms into Combinatory Logic. Adapting the presentation in [1]:152, [5] define $\langle\cdot\rangle$, which maps an arbitrary gapped structure into a NL_{CL} structure:

$$
\begin{aligned}
\langle x \rangle &\equiv x \\
\langle p{\cdot}q \rangle &\equiv \langle p \rangle {\cdot} \langle q \rangle \\
\langle p \circ q \rangle &\equiv \langle p \rangle \circ \langle q \rangle \\
\langle \lambda x.p \rangle &\equiv \mathbb{A}(x, \langle p \rangle) \\
\mathbb{A}(x, x) &\equiv \mathsf{I} \\
\mathbb{A}(x, p{\cdot}q) &\equiv (\mathsf{B}{\cdot}p){\cdot}\mathbb{A}(x, q) \quad (x \text{ not free in } p) \\
\mathbb{A}(x, p{\cdot}q) &\equiv (\mathsf{C}{\cdot}\mathbb{A}(x, p)){\cdot}q \quad (x \text{ not free in } q) \\
\mathbb{A}(x, x \circ q) &\equiv (\mathsf{C}{\cdot}\mathsf{I}) \circ q \quad (x \text{ not free in } q)
\end{aligned}
\tag{16}
$$

With this mapping defined, I can state the following three theorems given in [5]) characterizing the relationship between NL_λ and NL_{CL}:

Theorem (Faithfullness of the $\langle\cdot\rangle$ mapping from λ-structures into CL-structures): For any structure p and context $\Gamma\lceil\ \rceil$,

$$
\frac{\langle p \circ \lambda x \Gamma\lceil x \rceil \rangle}{\langle \Gamma\lceil p \rceil \rangle}\, CL \tag{17}
$$

Here, CL schematizes over some series of structural inferences allowable in NL_{CL}.

Theorem (Embedding of λ-free theorems of NL_λ in NL_{CL}): For any derivation in NL_λ (with abstraction restricted to $\lceil\ \rceil$-contexts) whose final sequent does not contain any λ-structures, there is an equivalent derivation in NL_{CL}.

Here, two derivations are equivalent if they differ only in the application of structural rules. They must have the same axiom instances, the same conclusion, and the Curry-Howard labeling must be the same up to α-equivalence.

Theorem (Embedding of IBC-free theorems of NL_{CL} in NL_λ): for any derivation in NL_{CL} whose conclusion does not contain the structures I, B, or C, there is an equivalent derivation in NL_λ.

The equivalence involves replacing each instance of I, B, and C with instances of the lambda postulate as follows:

$$
\begin{aligned}
&\frac{p}{p \circ \mathsf{I}}\, \mathsf{I} && \sim && \frac{p}{p \circ \lambda x x}\, \lambda \\
&\frac{p{\cdot}(q \circ r)}{q \circ ((\mathsf{B}{\cdot}p){\cdot}r)}\, \mathsf{B} && \sim && \frac{p{\cdot}(q \circ r)}{q \circ \lambda x (p{\cdot}(x \circ r))}\, \lambda \\
&\frac{(p \circ q){\cdot}r}{p \circ ((\mathsf{C}{\cdot}q){\cdot}r)}\, \mathsf{C} && \sim && \frac{(p \circ q){\cdot}r}{p \circ \lambda x ((x \circ q){\cdot}r)}\, \lambda
\end{aligned}
\tag{18}
$$

Note that each of these applications of the lambda postulate obeys the restriction to $\lceil\ \rceil$-contexts.

Thus NL_λ (with the lambda-postulate restricted to $\lceil\ \rceil$-contexts) and NL_{CL} are equivalent: any sequent containing only structures built from $\cdot$ and $\circ$ will be

a theorem of one just in case it is a theorem of the other. Furthermore, for each derivation in one system, there will be a matching derivation in the other that differs only in the application of structural rules, which means that the semantic values of the two derivations will be identical. Since NL_{CL} is conservative with respect to the non-associative Lambek grammar NL, NL_λ is too. As a result, NL_λ with restricted abstraction contexts can be used with full confidence that it is equivalent to an ordinary and well-behaved substructural grammar.

5 Cut elimination and decidability

5.1 Cut Elimination for NL_λ

The cut rule characterizes transitivity of the logical system:

$$\frac{\Gamma \vdash A \qquad \Sigma[A] \vdash B}{\Sigma[\Gamma] \vdash B}\ \text{CUT} \tag{19}$$

The cut rule says that if Γ is a proof of A, and Σ is a proof of B that depends on proving A, then we can construct a new proof of B in which A has been replaced with Γ. The formula A has been 'cut out' of the derivation.

The proof strategy, just as it was above for completeness, will be to rely on Restall's general proof of cut elimination for Gentzen-style sequent systems. This strategy emphasizes the ordinariness and the standardness of the logics here, and how they fit into a larger landscape of substructural logics.

In order for Restall's proof to apply, we need to demonstrate that the cut rule, the structural rule, and the logical rules conform to certain conditions. This is perfectly straightforward (see [5] for full details). Therefore we have:

Theorem (Cut Elimination): given that the parameter conditions, the eliminability of matching principal constituents, and the regularity condition hold, if $\Gamma \vdash A$ and $\Delta[A] \vdash B$ are provable, then $\Delta[\Gamma] \vdash B$ is also provable.

Proof: see [20]: Sect. 6.3.

5.2 Decidability of NL_λ

Decidability is a property a logic has if it is always possible to figure out whether a sequent is a theorem (has a proof, has a derivation) in a bounded amount of time, where the bound is some concrete function of the complexity of the sequent to be proved.

The structural postulate given above in (7) is a reversible inference, that is, it is bidirectional. In the discussion that follows, it will be helpful to keep track of the two directions separately:

$$\frac{\Sigma[\Delta\lceil A\rceil] \vdash B}{\Sigma[A \circ \lambda x \Delta\lceil x\rceil] \vdash B}\ \text{REDUCTION} \qquad \frac{\Sigma[A \circ \lambda x \Delta\lceil x\rceil] \vdash B}{\Sigma[\Delta\lceil A\rceil] \vdash B}\ \text{EXPANSION} \tag{20}$$

Since in proof search we are starting with the conclusion and trying to find appropriate premises, the names 'reduction' and 'expansion' are relative to the bottom-to-top direction of reading proofs. The main challenge for decidability is that there is no limit to the opportunities for expansion, since $B \equiv B \circ \lambda xx \equiv (B \circ \lambda xx) \circ \lambda xx \equiv ....$

I will leave a full account of the decidability of NL_λ for another occasion. Nevertheless, I will discuss a strategy that handles the vast majority of cases. The goal will be to push each Expansion inference upwards in the proof until one of two things happens: either it encounters a matching Reduction instance, in which case the two rules cancel each other out, and can be eliminated from the proof; or else the expansion is adjacent to a logical rule that introduces the focussed occurrence of $\circ$.

It turns out that the only candidate for such a logical rule is $/\!\!/L$.

$$\dfrac{\dfrac{\lambda x\Gamma\lceil x\rceil \vdash A \qquad \Sigma\lceil B\rceil \vdash C}{\Sigma[B /\!\!/ A \circ \lambda x\Gamma\lceil x\rceil] \vdash C}\,/\!\!/L}{\Sigma[\Gamma\lceil B /\!\!/ A\rceil] \vdash C}\,\textsc{exp} \quad \equiv \quad \dfrac{\lambda x\Gamma\lceil x\rceil \vdash A \qquad \Sigma[B] \vdash C}{\Sigma[\Gamma\lceil B /\!\!/ A\rceil] \vdash C}\,/\!\!/L_\lambda \tag{21}$$

We can replace the adjacent pair of inferences on the left with the derived inference on the right, which we can call $/\!\!/L_\lambda$. By repeated application of this reasoning, almost every instance of Expansion can either be eliminated, or replaced with an instance of $/\!\!/L_\lambda$. (There are exceptions that include certain parasitic scope configurations that will not be discussed here).

Having eliminated almost all expansion inferences, we can eliminate Reduction inferences in a similar fashion. That is, reasoning dually, Reduction inferences can be pushed *downwards* until the Reduction encounters an instance of $\backslash\!\!\backslash R$ that targets the $\circ$ connective introduced by Reduction. And once again, we can replace the combination of the reduction and the instance of $\backslash\!\!\backslash R$ with a derived rule that captures their net effect:

$$\dfrac{\dfrac{\Gamma\lceil A\rceil \vdash B}{A \circ \lambda x\Gamma\lceil x\rceil \vdash B}\,\textsc{red}}{\lambda x\Gamma\lceil x\rceil \vdash A \backslash\!\!\backslash B}\,\backslash\!\!\backslash R \quad \equiv \quad \dfrac{\Gamma\lceil A\rceil \vdash B}{\lambda x\Gamma\lceil x\rceil \vdash A \backslash\!\!\backslash B}\,\backslash\!\!\backslash R_\lambda \tag{22}$$

At this point, we have two derived logical inferences: $\backslash\!\!\backslash R_\lambda$, and $/\!\!/L_\lambda$. The $\backslash\!\!\backslash R_\lambda$ rule says that in-situ elements can take scope directly from embedded positions, without needing to first be abstracted leftwards. Dually, the $/\!\!/L_\lambda$ rule says that a context can surround a scope-taker even when the scope-taker is embedded in a still larger surrounding context. Adding the two derived logical rules to the standard logical rules leads to derivations of in-situ scope-taking, illustrated here for the sentence *Ann saw everyone*:

$$\dfrac{\dfrac{\dfrac{\mathrm{ann}\cdot(\mathrm{saw}\cdot\mathrm{DP}) \vdash \mathrm{S}}{\lambda x.\mathrm{ann}\cdot(\mathrm{saw}\cdot x) \vdash \mathrm{DP} \backslash\!\!\backslash \mathrm{S}}\,\backslash\!\!\backslash R_\lambda \qquad \mathrm{S} \vdash \mathrm{S}}{\mathrm{ann}\cdot(\mathrm{saw}\cdot\mathrm{S} /\!\!/ (\mathrm{DP} \backslash\!\!\backslash \mathrm{S})) \vdash \mathrm{S}}\,/\!\!/L_\lambda}{\mathrm{ann}\cdot(\mathrm{saw}\cdot\mathrm{everyone}) \vdash \mathrm{S}} \tag{23}$$

In effect, we have compiled both parts of the structural rule into the logical rules. If we add these two derived logical rules to the grammar, we can consider an approximation of NL_λ that consists entirely of logical rules.

Note that in this modified logic, each inference rule, including the derived inference rules, eliminates exactly one logical connective. As a result, no part of the proof can have a depth greater than the number of logical connectives in the final sequent. Since there is at most a finite number of ways to apply each rule to a given occurrence of a logical connective, decidability of the modified logic follows immediately.

5.3 Proof Search with Gaps

The treatment of scope-taking can be extended to a treatment of overt syntactic movement (see [5]). From the point of view of decidability, gaps are a challenge, since they allow us to posit new structure during the course of a proof search, in which case we lose the subformula property. An extension of the technique developed in the previous section allows derivations with gaps without giving up decidability.

$$\frac{\Gamma[B\cdot A] \vdash C}{\Gamma[A] \vdash B \backslash\!\backslash C} \backslash\!\backslash R_{lgap} \qquad \frac{\Gamma[A\cdot B] \vdash C}{\Gamma[A] \vdash B \backslash\!\backslash C} \backslash\!\backslash R_{rgap} \tag{24}$$

Since each of these inferences has the subformula property, and moreover, eliminates a logical connective, adding them to the logic will not compromise decidability.

To illustrate these logical rules in action, here is a derivation of the wh-question *Who did Ann see* (with *did* suppressed for simplicity):

$$\dfrac{\dfrac{\dfrac{\text{ann}\cdot(\text{see}\cdot\text{DP}) \vdash \text{S}}{\text{ann}\cdot\text{see} \vdash \text{DP} \backslash\!\backslash \text{S}} \backslash\!\backslash R_{lgap} \qquad \text{Q} \vdash \text{Q}}{\text{Q}/(\text{DP} \backslash\!\backslash \text{S})\cdot(\text{ann}\cdot\text{see}) \vdash \text{Q}} /L}{\text{who}\cdot(\text{ann}\cdot\text{see}) \vdash \text{Q}}\ \text{LEX} \tag{25}$$

6 Comparisons with Other Approaches

The participants at LENSL11 kindly suggested a number of other approaches to the logic of scope-taking that it would be useful to compare with the approach presented here. In this section, I will discuss Moortgat's [14] q type constructor (mentioned above); a multi-modal analysis also due to Moortgat [15]; Morrill et al.'s notion of scope-taking as discontinuity [16,18]; and, finally, standard Quantifier Raising.

6.1 Deriving the q Type Constructor

If we carry the strategy in Sect. 5 of fusing inferences into derived inferences one step further, we derive the rule of use for Moortgat's q type constructor, given above in (5):

$$\dfrac{\dfrac{\Gamma[A] \vdash B}{\lambda x \Gamma[x] \vdash A \backslash\!\backslash B}\backslash\!\backslash R_\lambda \qquad \Sigma[C] \vdash D}{\Sigma[\Gamma[C /\!/ (A \backslash\!\backslash B)]] \vdash D} /\!/ L_\lambda \quad \approx \quad \dfrac{\Gamma[A] \vdash B \qquad \Sigma[C] \vdash D}{\Sigma[\Gamma[q(A,B,C)]] \vdash D}\, q \tag{26}$$

We now have an explanation for why it was impossible to find a general right rule for the q type constructor: it is because the q inference represents the fusion of two logically distinct inferences, each with their own left and right rules.

In support of the usefulness of factoring the q into independent components, consider 'parasitic scope', a technique proposed in [2] to account for the scope-taking behavior of adjectives such as *same* and *different*. Parasitic scope requires the inferences to be interleaved in a way that cannot be duplicated by the q inference alone:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\text{the}\cdot(\text{N}/\text{N}\cdot\text{waiter}))\cdot(\text{served}\cdot\text{DP}) \vdash \text{S}}{\lambda x.(\text{the}\cdot(\text{N}/\text{N}\cdot\text{waiter}))\cdot(\text{served}\cdot x) \vdash \text{DP}\backslash\!\backslash\text{S}}\backslash\!\backslash R_\lambda}{\lambda y \lambda x.(\text{the}\cdot(y\cdot\text{waiter}))\cdot(\text{served}\cdot x) \vdash (\text{N}/\text{N})\backslash\!\backslash(\text{DP}\backslash\!\backslash\text{S})}\backslash\!\backslash R_\lambda \qquad \text{DP}\backslash\!\backslash\text{S} \vdash \text{DP}\backslash\!\backslash\text{S}}{\lambda x.(\text{the}\cdot((\text{DP}\backslash\!\backslash\text{S}) /\!/ ((\text{N}/\text{N})\backslash\!\backslash(\text{DP}\backslash\!\backslash\text{S}))\cdot\text{waiter}))\cdot(\text{served}\cdot x) \vdash \text{DP}\backslash\!\backslash\text{S}} /\!/ L_\lambda}{\lambda x.(\text{the}\cdot(\text{same}\cdot\text{waiter}))\cdot(\text{served}\cdot x) \vdash \text{DP}\backslash\!\backslash\text{S}}\,\text{LEX} \qquad \text{S} \vdash \text{S}}{(\text{the}\cdot(\text{same}\cdot\text{waiter}))\cdot(\text{served}\cdot\text{S} /\!/ (\text{DP}\backslash\!\backslash\text{S})) \vdash \text{S}} /\!/ L_\lambda}{(\text{the}\cdot(\text{same}\cdot\text{waiter}))\cdot(\text{served}\cdot\text{everyone}) \vdash \text{S}}\,\text{LEX} \tag{27}$$

Although the innermost pair of $/\!/L_\lambda$ and $\backslash\!\backslash R_\lambda$ can be fused into a single instance of the q inference, the outermost pair cannot.

6.2 Comparison with a Unary Modality Strategy

Moortgat, in [15], gives an analysis that at first glance is strikingly similar to NL_{CL}. The heart of the approach is a set of three structural postulates, lined up here in (29) underneath the corresponding NL_{CL} postulates.

$$\text{NL}_{CL}: \qquad \dfrac{p}{p \circ \mathsf{I}}\,\mathsf{I} \qquad \dfrac{p\cdot(q\circ r)}{q\circ((\mathsf{B}\cdot p)\cdot r)}\,\mathsf{B} \qquad \dfrac{(p\circ q)\cdot r}{p\circ((\mathsf{C}\cdot q)\cdot r)}\,\mathsf{C} \tag{28}$$

$$\text{Unary modalities}: \qquad \dfrac{p}{p \circ \mathsf{I}}\,\text{P0} \qquad \dfrac{p\cdot(q\circ r)}{q\circ\langle r\rangle(p\cdot r)}\,P2 \qquad \dfrac{(p\circ q)\cdot r}{p\circ\langle l\rangle(q\cdot r)}\,P1 \tag{29}$$

Both sets of postulates regulate the interaction of two binary modalities, $\cdot$ and $\circ$. The postulates in (29) make use in addition of two unary modalities, $\langle l\rangle$, and $\langle r\rangle$. (I've omitted a third unary modality, $\Diamond$, in order to emphasize the similarities between the approaches, and to simplify the discussion immediately below.) The presence or absence of the $\langle l\rangle$ and $\langle r\rangle$ modalities track the path between the in-situ position of a scope-taker and its scope position, very much like the structural punctuation marks B and C do in a NL_{CL} derivation.

The key difference between the two systems is that decorating a constituent with a unary modality blocks further abstraction from that constituent. As a result, the unary modalities are able to track at most one scope path at a time in the general case. In contrast, NL_{CL} allows multiple scope-takers to simultaneously share the same abstraction path without those paths getting confused. For example, note that in NL_{λ}, the structure $p{\cdot}(q{\cdot}s)$ is equivalent to $q \circ (s \circ \lambda y \lambda x (p{\cdot}(x{\cdot}y)))$. A derivation of the corresponding equivalence in NL_{CL} is given on the left:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{q \circ (s \circ ((\mathsf{B}{\cdot}(\mathsf{B}{\cdot}p)){\cdot}((\mathsf{B}{\cdot}(\mathsf{C}{\cdot}I)){\cdot}I)))}{q \circ ((\mathsf{B}{\cdot}p){\cdot}(s \circ ((\mathsf{B}{\cdot}(\mathsf{C}{\cdot}I)){\cdot}I)))}\,\mathsf{B}}{q \circ ((\mathsf{B}{\cdot}p){\cdot}((\mathsf{C}{\cdot}I){\cdot}(s \circ I)))}\,\mathsf{B}}{q \circ ((\mathsf{B}{\cdot}p){\cdot}((\mathsf{C}{\cdot}I){\cdot}s))}\,\mathsf{I}}{p{\cdot}(q \circ ((\mathsf{C}{\cdot}I){\cdot}s))}\,\mathsf{C}}{p{\cdot}((q \circ I){\cdot}s)}\,\mathsf{I}}{p{\cdot}(q{\cdot}s)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{q \circ \langle r \rangle (p{\cdot}\langle l \rangle (s \circ \langle r \rangle (I{\cdot}I)))}{q \circ \langle r \rangle (p{\cdot}\langle l \rangle (I{\cdot}(s \circ I)))}\,P2}{q \circ \langle r \rangle (p{\cdot}\langle l \rangle (I{\cdot}s))}\,P0}{p{\cdot}(q \circ \langle l \rangle (I{\cdot}s))}\,P2}{p{\cdot}((q \circ I){\cdot}s)}\,P1}{p{\cdot}(q{\cdot}s)}\,P0 \tag{30}$$

The derivation on the right using unary modalities can't be completed. The structure s gets trapped underneath an instance of the $\langle l \rangle$ operator, which prevents s from taking scope just underneath q.

This limitation prevents the unary modality strategy from accounting for the full range of scope analyses that have been proposed in the literature. In particular, the configuration derived by NL_{λ} and NL_{CL} in the example in (30) is an instance of parasitic scope. Parasitic scope has been advocated as a scope-taking strategy for handling a number of phenomena, including adjectives of comparison such as *same* and *different* [2], as illustrated in (27); respective and symmetrical predicates [12]; certain uses of the adjective *average* [9]; non-constituent coordination [11]; as well as for verb phrase ellipsis, sluicing, and anaphora in general [3], following [18]. NL_{λ} and NL_{CL} were originally proposed precisely in order to handle parasitic scope.

6.3 Comparison with Discontinuous Lambek Grammar

Morrill, Valentín and Fadda, in [18] and [21], present a type-logical grammar called Discontinuous Lambek Grammar. Though different from NL_{λ} in its historical development (see [2] versus [18]:11) and in form, the expressive power and the specific analyses it provides are closely parallel to those of NL_{λ}.

On a conceptual level, there is a dramatic difference. Discontinuous Lambek Grammar views the argument that a scope-taker combines with (its nuclear scope) as a discontinuous constituent. For instance, in the sentence *Mary claimed John wanted everyone to read the book*, the nuclear scope of *everyone* corresponds to the discontinuous string *John wanted ... to read the book*. Continuation-based grammars such as NL_{λ} and NL_{CL} view this portion of a linguistic tree as a unit: it is a

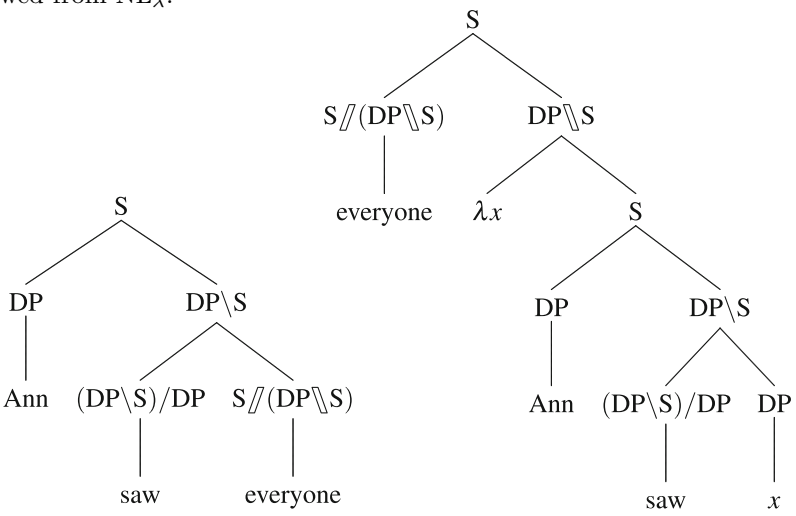
constituent with one piece removed (in the position of the scope-taker). All of its parts are connected, so it is a contiguous, single constituent, as illustrated in (8).

The correspondence between Discontinuous Lambek Grammar and NL_λ is easiest to see at the level of categories. Following [16], Morrill et al. define a type connective '↑' such that $B \uparrow A$ means (roughly) "a discontinuous expression that would be of category B if one of its gaps were filled with an expression of category A". This is functionally equivalent to our $A \backslash\!\backslash B$ (note the reversal of the order of the subcategories). Likewise, they define a complementary connective '↓' such that $D \downarrow C$ means "an expression that would be of category C, if only it were first substituted into a discontinuous expression of category D", which is functionally equivalent to our $C /\!/ D$. (Note again the reversal of the categories.) So their category for a generalized quantifier is $((\mathrm{S}_1 \uparrow \mathrm{DP}) \downarrow \mathrm{S}_2)$, which is equivalent to our $\mathrm{S}_2 /\!/ (\mathrm{DP} \backslash\!\backslash \mathrm{S}_1)$. As a result of this correspondence, multiple levels of discontinuity in Discontinuous Lambek Grammar can be handled as different varieties of parasitic scope in NL_λ and NL_{CL}, and vice versa.

In addition to a major difference in conceptual foundations, Morrill et al. are committed to the assumption that natural language is fully associative, that is, that the structures $p \cdot (q \cdot r)$ and $(p \cdot q) \cdot r$ are fully equivalent. Associativity is well-established as a default assumption in some varieties of categorial grammar. However, it is by no means clear that natural language is uniformly associative. Instead of building associativity into the basic definitions of the grammar, as Morrill et al. do, a more conservative strategy would be to build a non-associative grammar, and add associativity in a carefully regulated way, only where needed, as advocated in [17]. In that spirit, associativity can easily be added to NL_λ or NL_{CL} simply by adding an appropriate structural postulate, if desired.

6.4 Comparison with Quantifier Raising

Here is the structural operation of Quantifier Raising, illustrated with categories borrowed from NL_λ:



In NL_λ, this derivational step can be simulated closely using the λ postulate (reading the proof from bottom upwards):

$$\frac{\text{everyone} \circ \lambda x\,(\text{ann}\cdot(\text{saw}\cdot x)) \vdash \text{S}}{\text{ann}\cdot(\text{saw}\cdot\text{everyone}) \vdash \text{S}}\,\lambda \tag{31}$$

So the logic here captures a significant portion of the insight embodied in Quantifier Raising. Is NL_λ then just the logic of Quantifier Raising? In some sense, clearly yes.

However, there are important differences between Quantifier Raising and the λ postulate of NL_λ.

For one, the lambda postulate is bidirectional. This reflects the fact that the two structures it relates are fully equivalent logically: they denote the same object in the model. In contrast, in the treatment in, e.g., [8], the pre-QR structure does not have a denotation. Thus the main motivation for executing an instance of Quantifier Raising is to produce a new meaning. In contrast, the lambda postulate here is a structural rule. Like all structural rules, the effect of the rule on the Curry-Howard labeling is null (no change to the semantic labeling). Quantifier Raising is conceived of as a rule that has a semantic effect but no syntactic effect (it constitutes 'covert' movement); the lambda postulate here has a syntactic effect, but no semantic effect. Its role in the logic is purely to characterize the syntactic operation by which a delimited continuation combines with its functor (by being surrounding by it) or its argument (by surrounding it).

For another difference, Quantifier Raising can create unbound traces.

Unbound trace: [[some [friend [of everyone]]][called]]
$\Rightarrow$[everyone(λy[[some [friend [of y]]][called]])]
$\Rightarrow$[[some [friend [of y]]](λx[everyone($\lambda y.x$)][called])]

If QR targets the embedded quantifier *everyone* first, and then targets the originally enclosing quantifier *some friend of* _, the variable introduced by the QR of *everyone* (in this case, y) will end up unbound (free) in the final Logical Form structure. In a QR system, such derivations must be stipulated to be ill-formed. In the logics developed here, unbound traces cannot cause any problems.

Finally, although I have not emphasized this in the discussion here, the substructural logics given here allow fine-grained control over order of evaluation, allowing accounts of order-sensitive phenomena such as crossover, reconstruction, negative polarity licensing, and more. Evaluation order and its applications in natural language are discussed in detail in [5].

7 Conclusion

What is the logic of scope? With natural language in mind, here is my answer: when an expression takes scope, it combines with one of its delimited continuations. The substructural logics given here, NL_λ and NL_{CL}, illustrate two equivalent ways to implement a concrete continuation-based grammar. These grammars

are perfectly kosher substructural logics. In particular, they are sound and complete with respect to the usual class of models, they are conservative with respect to NL, and they enjoy cut elimination. Finally, although I presented a promising proof search strategy, a full account of decidability will have to wait for a future occasion.

References

1. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics. Elsevier, North-Holland (1984)
2. Barker, C.: Parasitic scope. Linguist. Philoso. **30**, 407–444 (2007)
3. Barker, C.: Scopability and sluicing. Linguist. Philosop. **36**(3), 187–223 (2013)
4. Barker, C.: Scope. In: Lappin, S., Fox, C. (eds.) The Handbook of Contemporary Semantics, 2d edn. Wiley-Blackwell, Malden (2014)
5. Barker, C.: Shan, Chung-chieh: Continuations and Natural Language. Oxford University Press, New York (2014)
6. Charlow, S.: On the semantics of exceptional scope. NYU Ph.D. dissertation (2014)
7. de Groote, P.: Towards abstract categorial grammars. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Morgan Kaufmann (2002)
8. Heim, I., Kratzer, A.: Semantics in generative grammar. Oxford (1998)
9. Kennedy, C., Stanley, J.: On average. Mind **118**(471), 583–646 (2009)
10. Kiselyov, O., Shan, C.: Continuation hierarchy and quantifier scope. In: McCready, E., Yabushita, K., Yoshimoto, K. (eds.) Formal Approaches to Semantics and Pragmatics. Springer, Netherlands (2014)
11. Kubota, Y.: Nonconstituent coordination in Japanese as constituent coordination: An analysis in Hybrid Type-Logical Categorial Grammar. Linguistic Inquiry 46.1 (2015)
12. Kubota, Y., Levine, R.: Unifying local and nonlocal modelling of respective and symmetrical predicates. In: Morrill, G., Muskens, R., Osswald, R., Richter, F. (eds.) Formal Grammar. LNCS, vol. 8612, pp. 104–120. Springer, Heidelberg (2014)
13. Lambek, J.: The mathematics of sentence structure. Am. Math. Monthly **60**(3), 154–170 (1958)
14. Moortgat, M.: Categorial Investigations: Logical and Linguistic aspects of the Lambek calculus. Foris, Dordrecht (1988)
15. Moortgat, M.: In situ binding: a modal analysis. In: Dekker, P., Stokhof, M. (eds.) Proceedings of the 10th Amsterdam Colloquium, pp. 539–549. Institute for Logic, Language and Computation, Universiteit van Amsterdam (1996a)
16. Moortgat, M.: Generalized quantification and discontinuous type constructors. In: Bunt, H.C., van Horck, A. (eds.) Discontinuous Constituency, pp. 181–207. Mouton de Gruyter, Berlin (1996b)
17. Moortgat, M.: Categorial Type Logics. In: van Benthem, J., ter Meulen, A.G.B. (eds.) The Handbook of Logic and Language. Elsevier, Amsterdam (1997)
18. Morrill, G., Valentn, O., Fadda, M.: The displacement calculus. J. Logic Lang. Inform. **20**(1), 1–48 (2011)
19. Muskens, R.: λ-grammars and the syntax-semantics interface. In: van Rooij, R., Stokhof, M. (eds) Proceedings of the 13th Amsterdam Colloquium, ILLC (2001)
20. Restall, G.: An introduction to substructural logics. Routledge, London (2000)
21. Valentín, O.: Theory of discontinuous lambek calculus. Ph.D. thesis, Universitat Autonma de Barcelona (2012)