Reasoning about contexts in Lambek Grammars

Chris Barker, *NYU* and Chung-chieh Shan, *RUTGERS*
chris.barker@nyu.edu and ccshan@rutgers.edu
http://homepages.nyu.edu/~cb125
http://www.cs.rutgers.edu/~ccshan

Version of October 16, 2007. **Partial rough draft—do not distribute.**

1. What is the logic of scope-taking in natural language?

The core example of scope-taking is, of course, in-situ quantification: in a sentence like *John saw everyone*, despite the fact that the quantifier *everyone* is embedded inside the verb phrase, it somehow ends up taking scope over the rest of the sentence. That is, the interpretation of *everyone* takes for its semantic argument a property representing the remainder of the clause: $\mathbf{everyone}(\lambda x.\mathbf{saw}\,x\,\mathbf{j})$.

From the point of view of resource-sensitive logics such as Lambek grammars, we can rephrase the question like this: what structural resources characterize natural language scope-taking? Associativity gives a first approximation. It is not sufficient, however, since it allows for quantifiers in peripheral but not in medial position (e.g., *John saw everyone* but not *John saw everyone yesterday*). The now-familiar story is told in, for instance, Moortgat (1997).

Moortgat's (1991, 1997) q type-constructor beautifully captures the interaction of an in-situ scope-taker with its surrounding context. Unfortunately, a complete understanding of the logical natural of q has been hard to come by. For instance, although it is easy to write a rule of use, a suitable rule of proof remains elusive. Likewise, there are several decompositions of q into a variety of modes governed by structural postulates, but they are all unsatisfactory in one way or another: either they require more than two modes (e.g., Morrill 1994), or else they require additional mechanisms to prevent commutativity from leaking into the default mode (e.g., Barker and Shan 2006).

Three active strands of research offer fresh insights. Bernardi and Moortgat (2007) use Lambek-Grishin grammar to calculate semanic contexts using (undelimited) continuations. However, the logic alone does not provide a complete semantic analysis of scope-taking. Therefore they augment their Lambek-Grishin proofs with a Continuation-Passing Style (CPS) transform. The combined Lambek-Grishin + CPS system provides enough expressive power to describe in-situ quantification. We can still wonder, then, how best to characterize scope-taking within a substructural logic without using supplementary mechanisms.

Morrill et al. (2007) offer Discontinuous Lambek Grammar. On their view, the essential notion that characterizes scope-taking is discontinuity. They extend Lambek grammar to allow expressions that have (an unbounded number of) points of discontinuity. An in-situ quantifier, then, combines with an expression that contains one point of discontinuity, namely, the position into which the quantifier will be inserted.

The third strand of research is our own, as reported in this paper. We describe two closely related substructural logics, NL_{CL} and NL_{λ}. In contrast with Morrill et al., we will emphasize the sense in which the material that surrounds an in-situ quantifier forms a coherent constituent. For us, as in Barker and Shan (2006) and Bernardi and Moortgat (2007), the crucial concept for scope-taking is a continuation. A (delimited) continuation is a (portion of) an expression's semantic context. One way of putting it is that the discontinuity approach concentrates on the syntactic aspect of scope-taking by describing operations in terms of mappings on strings (in the discontinuous case, sets of strings). We concentrate on combination at the level of semantic composition, combining values with their contexts.

Unlike Discontinuous Lambek Grammar, Lambek-Grishin grammar also relies heavily on continuations. However, it relies on undelimited continuations. We diagnose the use of undelimted instead of delimited continuations as the key element that prevents Lambek-Grishin grammar from providing a self-contained account of scope-taking. In fact, Barker (2004) conjectures that natural languages use only delimited continuations. This is an empirical question, of course, and there may very well be phenomena for which undelimited continuations provide a natural and elegant analysis (presupposition comes to mind as a possibly relevant case). We will argue below, however, that in-situ quantification clearly requires delimited continuations.

The approach here is closely related to the account in Barker and Shan (2006). In that paper, we present a type-logical grammar which explicitly provides access to delimited continuations. As mentioned above, the grammar in Barker and Shan required supplementary devices to prevent unwanted commutativity. The logics presented here do not have any commutativity problems. In fact, we prove that NL_{CL} is conservative over NL: any NL sequent provable in NL_{CL} is also a theorem of NL.

In addition to the commutativity issue, there are two significant differences between the system given there and the grammars below. The first is that the logics here allow nested contexts. As we will show, this enables us to provide reasonable analyses of certain scope-taking adjectives such as *same* and *different* based on the analysis in Barker (in press). Our nested contexts seem to be closely analogous to expressions in Morrill et al.'s system that contain multiple points of discontinuity. (The Lambek-Grishin approach doesn't seem to have an analog.)

The second difference between the system here and Barker and Shan (2006) is that Barker and Shan explicitly regulate order of evaluation. They argue that restrictions on order of evaluation are crucially needed in order to explain such phenomena as weak crossover, superiority, and linear order effects. We continue to believe that explicit reasoning about order of evaluation is important for a correct understanding of natural language; however, in this paper, we concentrate entirely on scope-taking independently of order-of-evaluation issues.

2. NL$_{CL}$

We define NL$_{CL}$, a non-associative Lambek grammar with two modes: $\backslash$, $\bullet$, $/$, the normal default mode, and $\backslash\!\!\backslash$, $\circ$, $/\!\!/$, the continuation mode. Three structural postulates govern the interaction between the two modes. These postulates can be viewed as incorporating two combinators, B and C (using Curry's traditional names). Thus one possible way to read the "CL" part of the name NL$_{CL}$ is Combinatory Logic. However, it would be just as appropriate to think of this logic as Continuation Logic, or perhaps Context Logic.

It is important to note that the postulates do not provide the full expressive power of arbitrary combinators (cf. XXX). As we will show, NL$_{CL}$ adds to NL structures equivalent only to strictly linear combinators.

In this section we give a soundness and completeness result, and prove that NL$_{CL}$ is conservative over NL.

Let $\mathcal{A}$ be a set of atomic formula symbols. In the examples given below, for instance, $\mathcal{A} = \{\text{DP}, \text{S}, ...\}$. We have two modes: a default mode ($\backslash, \times, /$) and a continuation mode ($\backslash\!\!\backslash, \otimes, /\!\!/$). Let $\mathcal{F} \supset \mathcal{A}$ be the smallest set of formulas such that for all $A, B \in \mathcal{F}$,

$$\mathcal{F} \quad \supset \quad \{\ A\backslash B\ ,\ A \times B\ ,\ B/A\ ,\ A\backslash\!\!\backslash B\ ,\ A \otimes B\ ,\ B/\!\!/A\ \}.$$

Let $\mathcal{S} \supset \mathcal{F}$ be the smallest set of structures such that for all $X, Y \in \mathcal{S}$,

$$\mathcal{S} \quad \supset \quad \{\ X \bullet Y\ ,\ X \circ Y\ ,\ \mathsf{I}\ ,\ \mathsf{B}\ ,\ \mathsf{C}\ \}.$$

Then the logical content of the default connectives are given by the following logical rules (Moortgat 1997:129):

$$\frac{\Gamma \vdash A \qquad \Sigma[B] \vdash C}{\Sigma[(\Gamma \bullet A \backslash B)] \vdash C} \backslash L \qquad \frac{A \bullet \Gamma \vdash C}{\Gamma \vdash A \backslash C} \backslash R$$

$$\frac{\Gamma[(A \bullet B)] \vdash C}{\Gamma[A \times B] \vdash C} \times L \qquad \frac{\Gamma \vdash A \qquad \Sigma \vdash B}{(\Gamma \bullet \Sigma) \vdash A \times B} \times R$$

$$\frac{\Gamma \vdash A \qquad \Sigma[B] \vdash C}{\Sigma[(B/A \bullet \Gamma)] \vdash C} /L \qquad \frac{\Gamma \bullet B \vdash C}{\Gamma \vdash C/B} /R$$

The logical content of the continuation mode connectives is identical, up to substituting $\backslash\!\!\backslash$, $\otimes$ and $/\!\!/$ for $\backslash$, $\times$ and $/$, and substituting the structural connective $\circ$ for $\bullet$:

$$\frac{\Gamma \vdash A \qquad \Sigma[B] \vdash C}{\Sigma[(\Gamma \circ A \backslash\!\!\backslash B)] \vdash C} \backslash\!\!\backslash L \qquad \frac{A \circ \Gamma \vdash C}{\Gamma \vdash A \backslash\!\!\backslash C} \backslash\!\!\backslash R$$

$$\frac{\Gamma[(A \circ B)] \vdash C}{\Gamma[A \otimes B] \vdash C} \times L \qquad \frac{\Gamma \vdash A \qquad \Sigma \vdash B}{(\Gamma \circ \Sigma) \vdash A \otimes B} \times R$$

$$\frac{\Gamma \vdash A \qquad \Sigma[B] \vdash C}{\Sigma[(B /\!\!/ A \circ \Gamma)] \vdash C} /\!\!/ L \qquad \frac{\Gamma \circ B \vdash C}{\Gamma \vdash C /\!\!/ B} /\!\!/ R$$

Note that we use $\bullet$ as the structural analog of the fusion connective $\times$ in the default mode, and likewise $\circ$ as the structural analog of the fusion connectives $\otimes$ in the continuation mode.

These logical rules are completely ordinary for multi-modal Lambek grammars, and have been copied verbatim from Moortgat (1997:129). In addition, the Curry-Howard labelling is likewise completely standard: left rules correspond to unpairing for the fusion operators and function application for the adjoints, and the right rules correspond to pairing for the fusion connectives and lambda-abstraction for the adjoints. See Moortgat (1997:120) for complete details on the Curry-Howard labeling.

NL$_{CL}$ has three structural postulates:

$$\frac{p}{p \circ \mathsf{I}} \,\mathsf{I}$$

$$\frac{p \bullet (q \circ r)}{q \circ ((\mathsf{B} \bullet p) \bullet r)} \,\mathsf{B}$$

$$\frac{(p \circ q) \bullet r}{p \circ ((\mathsf{C} \bullet q) \bullet r)}\ \mathsf{C}$$

Restall (2000) considers I (which he writes '0') as (p. 30) "a zero-place punctuation mark," where punctuation marks (p. 19) "stand to structures in the same way that connectives stand to formulae." 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 equally valid. Restall calls the top-to-bottom inference for the identity postulate Push, and the other direction Pop. [1]

I is a right identity with respect to $\circ$, of course. In some sense, B governs mixed commutativity involving $\bullet$ and $\circ$, and C governs mixed associativity involving $\bullet$ and $\circ$. However, we will suggest below a number of other ways of understanding what these postulates are doing.

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

$$
\begin{prooftree}
\AxiomC{$\textsc{dp} \vdash \textsc{dp}$}
\AxiomC{$\textsc{s} \vdash \textsc{s}$}
\RightLabel{$\backslash L$}
\BinaryInfC{$\textsc{dp} \bullet \textsc{dp}\backslash\textsc{s} \vdash \textsc{s}$}
\RightLabel{$/L$}
\UnaryInfC{$\textsc{dp} \bullet ((\textsc{dp}\backslash\textsc{s})/\textsc{dp} \bullet \textsc{dp}) \vdash \textsc{s}$}
\RightLabel{LEX}
\UnaryInfC{$\textit{John} \bullet (\textit{saw} \bullet \textsc{dp}) \vdash \textsc{s}$}
\RightLabel{I}
\UnaryInfC{$\textit{John} \bullet (\textit{saw} \bullet (\textsc{dp} \circ \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{B}
\UnaryInfC{$\textit{John} \bullet (\textsc{dp} \circ ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{B}
\UnaryInfC{$\textsc{dp} \circ ((\mathsf{B} \bullet \textit{John}) \bullet ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{$\backslash\!\!\backslash R$}
\UnaryInfC{$(\mathsf{B} \bullet \textit{John}) \bullet ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I}) \vdash \textsc{dp}\backslash\!\!\backslash\textsc{s}$}
\AxiomC{$\textsc{s} \vdash \textsc{s}$}
\RightLabel{$/\!\!/L$}
\BinaryInfC{$\textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s}) \circ ((\mathsf{B} \bullet \textit{John}) \bullet ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{LEX}
\UnaryInfC{$\textit{everyone} \circ ((\mathsf{B} \bullet \textit{John}) \bullet ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{B}
\UnaryInfC{$\textit{John} \bullet (\textit{everyone} \circ ((\mathsf{B} \bullet \textit{saw}) \bullet \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{B}
\UnaryInfC{$\textit{John} \bullet (\textit{saw} \bullet (\textit{everyone} \circ \mathsf{I})) \vdash \textsc{s}$}
\RightLabel{I}
\UnaryInfC{$\textit{John} \bullet (\textit{saw} \bullet \textit{everyone}) \vdash \textsc{s}$}
\end{prooftree}
$$

The Curry-Howard labeling for this derivation is

$$\mathbf{everyone}(\lambda x.\mathbf{saw}\, x\, \mathbf{j})$$

[1]In the form of an official inference rule, the identity postulate is written $\dfrac{\Sigma[\Gamma[p]] \vdash A}{\Sigma[p \circ \lambda x \Gamma[x]] \vdash A}$, and similarly for the other two rules.

It should be clear that we have the usual two classes of derivations for *Someone saw everyone*, one class giving linear scope, and the other class giving inverse scope, as well as derivations with unbounded long distance scope-taking (e.g., *Someone asked everyone to leave*).

We will provide derivations of some more interesting examples below, after first establishing some general properties of NL$_{CL}$.

2.1. Soundness and completeness.
NL$_{CL}$ is sound and complete with respect to the usual class of relational models. We will give enough details to make it clear that soundness and completeness follows directly from the proofs given in Restall (2000 chapter 11).

A frame $\mathcal{F}$ for NL$_{CL}$ consists of

- A (flat) set of points $\mathcal{P}$
- 3-place accessibility relations $R_\bullet$ and $R_\circ$
- 1-place predicates I, B, and C

A model $\mathfrak{M}$ for NL$_{CL}$ is a frame along with an evaluation relation $\Vdash$ that satisfies the following:

$$x \Vdash B/A \text{ iff } \forall y,z.(R_\bullet xyz \wedge y \Vdash A) \rightarrow z \Vdash B$$
$$y \Vdash A\backslash B \text{ iff } \forall x,z.(R_\bullet xyz \wedge x \Vdash A) \rightarrow z \Vdash B$$
$$z \Vdash A \times B \text{ iff } \exists x,y.R_\bullet xyz \wedge x \Vdash A \wedge y \Vdash B$$

$$x \Vdash B /\!\!/ A \text{ iff } \forall y,z.(R_\circ xyz \wedge y \Vdash A) \rightarrow z \Vdash B$$
$$y \Vdash A \backslash\!\!\backslash B \text{ iff } \forall x,z.(R_\circ xyz \wedge x \Vdash A) \rightarrow z \Vdash B$$
$$z \Vdash A \otimes B \text{ iff } \exists x,y.R_\circ xyz \wedge x \Vdash A \wedge y \Vdash B$$

$$x \Vdash \mathsf{I} \text{ iff } x \in I$$
$$x \Vdash \mathsf{B} \text{ iff } x \in B$$
$$x \Vdash \mathsf{C} \text{ iff } x \in C$$
$$z \Vdash p \bullet q \text{ iff } \exists x,y.R_\bullet xyz \wedge x \Vdash p \wedge y \Vdash q$$
$$z \Vdash p \circ q \text{ iff } \exists x,y.R_\circ xyz \wedge x \Vdash p \wedge y \Vdash q$$

Restall (2000:249) provides an algorithm for constructing frame conditions corresponding to the structural postulates. Following Restall, we define a recursive function F mapping structural postulates to expressions in the first-order predicate calculus. For propositional variables, $F(p) = (p = x)$.

For structural connectives, we have

- $F(\mathsf{I}) = Ix$
- $F(\mathsf{B}) = Bx$
- $F(\mathsf{C}) = Cx$
- $F(X \bullet Y) = \exists y \exists z.R_\bullet yzx \wedge F(X)[x := y] \wedge F(Y)[x := z]$
- $F(X \circ Y) = \exists y \exists z.R_\circ yzx \wedge F(X)[x := y] \wedge F(Y)[x := z]$

Then for a structural rule $P = \dfrac{\Sigma[\Gamma] \vdash A}{\Sigma[\Gamma'] \vdash A}$ in which Γ and Γ' contain $p_1, p_2, ..., p_n$ as propositional variables,

$$F(P) = \forall x, p_1, p_2, ..., p_n.F(\Gamma') \to F(\Gamma)$$

For example, Push gives rise to the following frame condition:

$$F(\frac{p}{p \circ \mathsf{I}}) =$$

$$\forall xp.(\exists y \exists z R_\circ yzx \wedge F(p)[x := y] \wedge F(\mathsf{I})[x := z]) \to (p = x)$$

$$= \forall xp.(\exists y \exists z R_\circ yzx \wedge (p = x)[x := y] \wedge F(\mathsf{I})[x := z]) \to (p = x)$$

$$= \forall xp.(\exists y \exists z R_\circ yzx \wedge (p = y) \wedge F(\mathsf{I})[x := z]) \to (p = x)$$

$$= \forall xp.(\exists y \exists z R_\circ yzx \wedge (p = y) \wedge Ix[x := z]) \to (p = x)$$

$$= \forall xp.(\exists y \exists z R_\circ yzx \wedge (p = y) \wedge Iz) \to (p = x)$$

$$= \forall xp.(\exists z R_\circ pzx \wedge Iz) \to (p = x)$$

We adopting the following abbreviations in the style of Restall:

Structural rules:

$$\frac{\Sigma[\Gamma] \vdash A}{\Sigma[\Gamma'] \vdash A} \qquad \equiv \qquad \frac{\Gamma}{\Gamma'}$$

Implicit universals:

$$Rxyz \qquad \equiv \qquad \forall x, y, z.Rxyz$$

Implicit existentials, one-place:

$$Rx(T)z \qquad \equiv \qquad \exists y.Rxyz \wedge Ty$$

Implicit existentials, three-place:

$$R_1x(R_2uv)z \qquad \equiv \qquad \exists y.R_1xyz \wedge R_2uvy$$

This gives the following set of frame conditions:

Structural postulate:	Frame condition:
$\dfrac{p}{p \circ \mathsf{I}}$	$R_\circ x(I)y \leftrightarrow x = y$
$\dfrac{p \bullet (q \circ r)}{q \circ ((\mathsf{B} \bullet p) \bullet r)}$	$R_\circ q(R_\bullet(R_\bullet(B)p)r)x \leftrightarrow R_\bullet p(R_\circ qr)x$
$\dfrac{(p \circ q) \bullet r}{p \circ ((\mathsf{C} \bullet q) \bullet r)}$	$R_\circ p(R_\bullet(R_\bullet(C)q)r)x \leftrightarrow R_\bullet(R_\circ pq)rx$

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

Proof: given in Restall (2000), theorems 11.20, 11.37.

2.2. NL$_{CL}$ is conservative over NL.

Because in-situ scope-taking requires establishing a long-distance dependency with medial embedded elements, it requires both associativity and commutativity. It is worth wondering whether commutativity could leak into the default mode, in effect allowing illicit scrambling of argument positions. For instance, in Barker and Shan (2006), the simplest version of their structural postulates gives rise to commutativity in the presence of a right identity (see Barker and Shan 2006:footnote 2). Barker and Shan eventually adopt more specific postulates for independent reasons in a way that fortuitously blocks unwanted commutativity. In any case, it is worth noting here that NL$_{CL}$ does not have any commutativity problem even its simplest, most general presentation.

In order to understand informally how NL$_{CL}$ avoids importing commutativity into the default mode, consider the point in a proof at which Push applies. Each instance of Push introduces two new structural connectives, $\circ$ and I. Imagine adding matching indexes to these new elements, thus: $\dfrac{p}{p \circ_i \mathsf{I}_i}$ Push. Later in the derivation, we will have to eliminate all occurrences of $\circ$. This can only be accomplished by an application of Pop. At the moment at which we are about to use Pop, we check to see whether the $\circ$ and the I we are about to eliminate have the same index. Unwanted commutativity in the default mode arises when Pop eliminates connectives that entered the derivation in different steps, i.e., that do not share an index. NL$_{CL}$ guarantees that Pop will eliminate occurrences of $\circ$ and I only if they were introduced by the same application of Push. The way that NL_{CL} manages to maintain this discipline is by adding a bookkeeping mechanism in the form of the structural connectives B and C. The presence of these elements forces the application of a series of structural postulates to unwind in the exact opposite order in which they were originally applied.

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

Proof: The if direction is easy, since every NL derivation is also an NL_{CL} derivation. In the only-if direction, we need to show that if ϕ is an NL sequent that is not valid in NL, it is not valid in NL_{CL}.

The proof proceeds by extending a falsifying NL model to a falsifying NL_{CL} model. Suppose that an NL sequent ϕ is not valid in NL. Then there is a model $\mathcal{M}$ of NL, whose (flat) point-set is $\mathcal{P}$ and whose (only) ternary relation is R, such that there is some $x \in \mathcal{P}$ that falsifies ϕ.

We want to extend the model $\mathcal{M}$ of NL to a new model $\mathcal{M}_{CL}$ of NL_{CL} in which the same point x still falsifies ϕ. Without loss of generality, assume $\mathcal{P}$ contains no ordered pairs. Let a, i, b, and c be four distinct points not in $\mathcal{P}$ that are also not ordered pairs. Let the (flat) point-set $\mathcal{P}_{CL}$ of the new model be the smallest set such that

- $\mathcal{P}$ is a subset of $\mathcal{P}_{CL}$;
- a, i, b, c are four elements of $\mathcal{P}_{CL}$;
- the ordered pair $\langle x, y\rangle$ is in $\mathcal{P}_{CL}$ whenever both x and y are in $\mathcal{P}_{CL}$ and either x or y is not in $\mathcal{P} \cup \{a\}$.

As one might expect, we let i be the (only) point in the new model that satisfies the predicate I, and likewise for b and c with respect to the predicates B and C.

Let $R_\bullet$ and $R_\circ$, the three-place relations of $\mathcal{M}_{CL}$, be the smallest relations such that

- $R_\bullet xyz$ if $Rxyz$, for any $x, y, z \in \mathcal{P}$.
- $R_\bullet xy\langle x, y\rangle$ if either x or y is not in $\mathcal{P} \cup \{a\}$, for any $x, y \in \mathcal{P}_{CL}$.
- $R_\bullet aaa$.
- $R_\circ xix$, for any $x \in \mathcal{P}_{CL}$.
- $R_\circ y\langle\langle b, x\rangle, z\rangle t$ if $R_\circ yzs$ and $R_\bullet xst$, for any $x, y, z, s, t \in \mathcal{P}_{CL}$.
- $R_\circ x\langle\langle c, y\rangle, z\rangle t$ if $R_\circ xys$ and $R_\bullet szt$, for any $x, y, z, s, t \in \mathcal{P}_{CL}$.

Note that $R_\bullet$ restricted to $\mathcal{P}$ coincides with R.

The recursive definition of $R_\circ$ is well-founded because the second argument to $R_\circ$ after "if" is always a proper subpart of the argument before "if". It is easy to check that $R_\bullet$ and $R_\circ$ satisfy the frame conditions. For instance, the frame condition for the right identity is precisely that $R_\circ xiy$ iff $x = y$.

Let each atomic proposition's valuation in the new model be the union of $\{a\}$ and its valuation in the old model. In other words, if X is an atomic proposition in NL and $x \in \mathcal{P}_{CL}$, then let x satisfy X in the new model $\mathcal{M}_{CL}$ iff $x = a$ or x satisfies X in the old model $\mathcal{M}$. It now remains to extend this property from atomic propositions X to all structures and formulas X, by induction on X. The inductive steps are $\bullet$, $\times$, $/$, and $\backslash$. Write $\Vdash$ for the satisfaction relation in the old model $\mathcal{M}$, and $\Vdash_{CL}$ for the satisfaction relation in the new model $\mathcal{M}_{CL}$.

If $X = X_1 \bullet X_2$, then we reason:

$$
\begin{array}{lll}
 & x \Vdash_{CL} X_1 \bullet X_2 & \\
\text{iff} & \exists y, z \in P_{CL} : y \Vdash_{CL} X_1 \wedge z \Vdash_{CL} X_2 \wedge R_\bullet yzx & \text{(by the definition of } \Vdash_{CL}\text{)} \\
\text{iff} & R_\bullet aax \vee (\exists y \in P : y \Vdash X_1 \wedge R_\bullet yax) & \text{(by induction hypothesis)} \\
 & \qquad \vee (\exists z \in P : z \Vdash X_1 \wedge R_\bullet azx) & \\
 & \qquad \vee (\exists y, z \in P : y \Vdash X_1 \wedge z \Vdash X_2 \wedge R_\bullet yzx) & \\
\text{iff} & x = a \vee \exists y, z \in P : y \Vdash X_1 \wedge z \Vdash X_2 \wedge Ryzx & \text{(by definition of } R_\bullet\text{)} \\
\text{iff} & x = a \vee x \Vdash X_1 \bullet X_2 & \text{(by definition of } \Vdash\text{)}
\end{array}
$$

The case for $\times$ is similar, except that X_1 and X_2 are restricted to formulas.

If $X = X_1/X_2$, then we reason:

$$
\begin{array}{lll}
 & x \Vdash_{CL} X_1/X_2 & \\
\text{iff} & \forall z \in \mathcal{P}_{CL} : (\exists y \in \mathcal{P}_{CL} : y \Vdash_{CL} X_2 \wedge R_\bullet xyz) & \text{(by definition of } \Vdash_{CL}\text{)} \\
 & \qquad\qquad \rightarrow z \Vdash_{CL} X_1 & \\
\text{iff} & \forall z \in \mathcal{P}_{CL} : (R_\bullet xaz \vee \exists y \in \mathcal{P} : y \Vdash X_2 \wedge R_\bullet xyz) & \text{(by induction hypothesis)} \\
 & \qquad\qquad \rightarrow (z = a \vee z \Vdash X_1) &
\end{array}
$$

Now, either $x \in \mathcal{P}$ or $x \notin \mathcal{P}$. If $x \in \mathcal{P}$, then the definition of $R_\bullet$ reduces the last proposition above to $\forall z \in \mathcal{P} : (\exists y \in \mathcal{P} : y \Vdash X_2 \wedge Rxyz) \rightarrow z \Vdash X_1$, which is equivalent to $x \Vdash X_1/X_2$. If $x \notin \mathcal{P}$, then the definition of $R_\bullet$ reduces the last proposition above to $x = a$. Hence $x \Vdash_{CL} X$ iff $x = a \vee x \Vdash X$, as desired. The $\backslash$ case is like the $/$ case.

Thus if x falsifies a sequent ϕ in NL, the same point x falsifies ϕ in NL$_{CL}$. Since NL$_{CL}$ is sound, ϕ is not provable in NL$_{CL}$. QED.

3. NL$_\lambda$

The postulates of NL$_{CL}$ were not chosen ad-hoc by trial and error. Rather, they are designed to implement Shönfinkel's embedding of (in this case, linear) λ-terms into combinatory logic. In fact, we will argue that scope-taking is a kind of hypothetical reasoning, and, like hypothetical reasoning, intimately related to linear lambda abstraction in a way that we will shortly make precise.

Adapting the presentation in Barendregt (1984:152), we define $\langle\cdot\rangle$, which maps an arbitrary λ-term into combinatory logic:

$$\begin{array}{rclrcl} \langle x\rangle & \equiv & x & \mathbb{A}(x,x) & \equiv & \textsc{i} \\ \langle MN\rangle & \equiv & \langle M\rangle\,\langle N\rangle & \mathbb{A}(x,M) & \equiv & \textsc{k}M \quad (x \text{ not free in } M) \\ \langle \lambda x.M\rangle & \equiv & \mathbb{A}(x,\langle M\rangle) & \mathbb{A}(x,MN) & \equiv & \textsc{s}(\mathbb{A}(x,M))(\mathbb{A}(x,N)) \end{array}$$

where $\textsc{s}xyz \rightsquigarrow_{CL} xz(yz)$, $\textsc{k}xy \rightsquigarrow_{CL} x$, and $\textsc{i}x \rightsquigarrow_{CL} x$ as usual. As Barendregt shows, if $M \rightsquigarrow_{\beta} N$, then $\langle M\rangle \rightsquigarrow_{CL} \langle N\rangle$.

For example,

$$\langle \lambda x \lambda y.yx\rangle = \textsc{s}(\textsc{k}(\textsc{si}))(\textsc{s}(\textsc{kk})\textsc{i})$$

We could use postulates that implement Shönkinkel's mapping verbatim. However, the frame conditions and the conservativity proof are simpler if we use instead a more efficient refinement of the mapping due to David Turner. We add the following clauses:

$$\begin{array}{rcll} \mathbb{A}(x,MN) & \equiv & \textsc{b}M(\mathbb{A}(x,N)) & (x \text{ not free in } M) \\ \mathbb{A}(x,MN) & \equiv & \textsc{c}(\mathbb{A}(x,M))N & (x \text{ not free in } N) \end{array}$$

where $\textsc{b}xyz \rightsquigarrow_{CL} x(yz)$ and $\textsc{c}xyz \rightsquigarrow_{CL} xzy$. Now

$$\langle \lambda x \lambda y.yx\rangle = \textsc{b}(\textsc{ci})\textsc{i}$$

Since the lambda-terms we have in mind will be linear, we can adapt Shönfinkel's mapping for NL$_{CL}$ (which we continue to write $\langle\cdot\rangle$) using only the $\textsc{b}$ and $\textsc{c}$ clauses:

$$\begin{array}{rcl} \langle x\rangle & \equiv & x \\ \langle p \bullet q\rangle & \equiv & \langle p\rangle \bullet \langle q\rangle \\ \langle \lambda x.p\rangle & \equiv & \mathbb{A}(x,\langle p\rangle) \end{array}$$

$$\begin{array}{rcll} \mathbb{A}(x,x) & \equiv & \textsc{i} & \\ \mathbb{A}(x,p\bullet q) & \equiv & (\textsc{b} \bullet p) \bullet \mathbb{A}(x,q) & (x \text{ not free in } p) \\ \mathbb{A}(x,p\bullet q) & \equiv & (\textsc{c} \bullet \mathbb{A}(x,p)) \bullet q & (x \text{ not free in } q) \end{array}$$

Note that the last three clauses of the mapping correspond closely to the three structural postulates of NL$_{CL}$.

This mapping allows us to define the following derived inference rule:

$$\frac{\Sigma[\Gamma\lceil p\rceil] \vdash A}{\Sigma[p \circ \langle \lambda x.\Gamma\lceil x\rceil\rangle] \vdash A}\ \lambda$$

We will illustrate this inference shortly, but first we must provide some details. As usual with λ, we pay for conceptual simplicity with some definitional complexity.

$$\Gamma\lceil p\rceil ::= \quad p \quad | \quad \lambda y.\,\Gamma\lceil p\rceil \quad | \quad q \bullet \Gamma\lceil p\rceil \quad | \quad \Gamma\lceil p\rceil \bullet q$$

In other words, this λ "abstracts" only over structures built from $\bullet$ and λ.

Thus the following inferences are licensed by the rule λ:

$$\frac{A}{A \circ \lambda x.x} \qquad \frac{A \bullet B}{A \circ \lambda x.(x \bullet B)} \qquad \frac{\lambda x.(x \bullet B)}{B \circ \lambda y \lambda x.(x \bullet y)}$$

But not these:

$$\frac{A \circ B}{A \circ \lambda x.(x \circ B)} \qquad \frac{\lambda x.(x \bullet B)}{B \circ \lambda x \lambda y.(x \bullet y)}$$

The reason these instances are not allowed is that abstraction across $\circ$ is forbidden (though see section X.Y below); and, as usual, the first argument of the lambda expression must correspond to the outermost λ.

When applying the inference rule, it is crucial that linearity be maintained. In practice, this means that each time the inference rule is applied, x must be chosen fresh, i.e., distinct from every other symbol in Γ.

Here is an example derivation of *John saw everyone* using the derived rule λ (compare with the derivation using the postulates given above in section 2):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\textsc{dp} \bullet ((\textsc{dp}\backslash\textsc{s})/\textsc{dp} \bullet \textsc{dp}) \vdash \textsc{s}}}{\textit{John} \bullet (\textit{saw} \bullet \textsc{dp}) \vdash \textsc{s}}\,\textsc{lex}}{\textsc{dp} \circ \lambda x(\textit{John} \bullet (\textit{saw} \bullet x)) \vdash \textsc{s}}\,\lambda}{\lambda x(\textit{John} \bullet (\textit{saw} \bullet x)) \vdash \textsc{dp}\backslash\!\!\backslash\textsc{s}}\,\backslash\!\!\backslash R \quad \textsc{s} \vdash \textsc{s}}{\dfrac{\textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s}) \circ \lambda x(\textit{John} \bullet (\textit{saw} \bullet x)) \vdash \textsc{s}}{\textit{John} \bullet (\textit{saw} \bullet \textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s})) \vdash \textsc{s}}\,\lambda}\,/\!\!/L$$

$$\textbf{everyone}(\lambda x.\textbf{saw}\, x\, \textbf{j})$$

We will often omit the angle brackets ($\langle\cdot\rangle$) for the sake of readability.

4. Delimited continuations

Barker and Shan (2006) call their $\backslash$, $\times$, $/$ mode the default mode, and their $\backslash\!\!\backslash$, $\otimes$, $/\!\!/$ mode the continuation mode. The default mode (i.e., the standard mode in a single-moded Lambek grammar) characterizes direction-sensitive function/argument combination. Thus the default mode fusion connective ('$\times$') expresses syntactic combination, and the residuated adjoints $/$ and $\backslash$ express left and right implication. This situation can be summarized with the usual residuation equivalences:

$$A \vdash C/B \qquad \text{iff} \qquad A \bullet B \vdash C \qquad \text{iff} \qquad B \vdash A\backslash C$$

Reading from left to right, if an A is the kind of expression that can combine with a B to its right to form an expression of category C, then the concatenation of A with B (in that order) yields an expression of category C, in which case a B alone is the kind of thing that can combine with an A to its left to form an expression of category C.

Barker and Shan (2006) view their continuation-mode fusion $\otimes$ not as expressing syntactic concatenation, but rather the combination of a plug with its context. They give a geometric interpretation of their structural postulates as a means of rotating through different ways of articulating a structure into a plug versus context. As we will discuss below, (delimited) continuations are essentially (partial) contexts, which is the motivation for calling this mode the continuation mode.

We will adopt a similar perspective here. The derived lambda inference makes clear the relationship between a plug and its context:

$$\frac{\Gamma\lceil p \rceil \vdash B}{p \circ \lambda x \Gamma\lceil x \rceil \vdash B}\,\lambda$$

We can continue this reasoning by considering the case where p is a formula A, in which case we have

$$\Gamma\lceil A \rceil \vdash B \qquad \text{iff} \qquad A \circ \lambda x \Gamma\lceil x \rceil \vdash B \qquad \text{iff} \qquad \lambda x \Gamma\lceil x \rceil \vdash A\backslash\!\!\backslash B$$

Thus structures that derive formulas of the form $A\backslash\!\!\backslash B$ are contexts (delimited continuations). Crucially, $A \circ B$ is not interpreted as concatenation, but as inserting A into the hole in the context B, wherever on the frontier of B that hole happens to be. (Below we discuss Morrill et al.'s approach, on which contexts are conceived as discontinuous constituents.)

We can now understand what the residuation equations are telling us:

$$A \vdash C /\!\!/ B \qquad \text{iff} \qquad A \circ B \vdash C \qquad \text{iff} \qquad B \vdash A \backslash\!\!\backslash C$$

The middle and right are clear: if A plugged into the context B forms something of category C (middle), then a B by itself must be the kind of context that requires an A-type plug to form a C. On the left, then, an A is the kind of expression that can plug into a B context to form a C.

Thus the default mode can be thought of as left/right concatenation, i.e., horizontal combination. The continuation mode can be thought of as plug/context combination, i.e., vertical combination.

Note that there is an inherent asymmetry in the continuation mode that is not present in the concatenation mode. Because the right hand element in a $\circ$ pair must be a context, it will have both an input category (the category of the plug) and an output (the result of plugging the plug into its context). This means in practice that the B formula will always be complex, of the form of a continuation (i.e., $A \backslash\!\!\backslash C$ for some choice of A and of C).

Natural language requires delimited, not undelimited, continuations. Bernardi and Moortgat (2007) offer an analysis of natural language scope-taking that is also based on continuations. Unlike the continuations in Barker and Shan (2006) and the continuations here, the continuations in their Lambek-Grishin grammars are undelimited. Very roughly, a delimited continuation is one that has both an input type and a result type (e.g., in the formula $A \backslash\!\!\backslash B$, A is the input type, the type of the needed plug, and B is the result type, the type of the result after plugging with A). An undelimited continuation is one whose type can be thought of as a negated type, $\neg A$, usually treated as $A \rightarrow \bot$.

In computational terms, a delimited continuation $A \backslash\!\!\backslash B$ is a function from objects of type A to objects of type B. An undelimited continuation $A \rightarrow \bot$ is a function on objects of type A that never returns.

Historically, undelimited continuations were studied first. They are well suited for describing the difference between intuitionistic reasoning and classical reasoning (Griffin, Parigot, de Groote). However, Barker (2004) conjectures that although many natural language constructions require an analysis in terms of delimited continuations, no natural language construction requires undelimited continuations.

It is well known that undelimited continuations can be used to simulate delimited continuations, as long as it is possible to store undelimited continuations for later retrieval. (The idea is that the stored continuation keeps track of what the delimited result type would have been.)

Lambek/Grishin grammars do not lend themselves to storage and retrieval, so Bernardi and Moortgat adopt a different strategy. Like de Groote (2001b), Barker (2002), and others, they equate interpret $\bot$ with a special result type, the type of a complete sentence. Lambek/Grishin proofs are mapped via an extended Curry-Howard correspondence into a term language that deals in undelimited continuations. (More specifically, Curien and Herblein's lambda mu co-mu calculus, in which undelimited continuations correspond to various flavors of co-implication.) Rather than computing with terms in this language directly, they map lambda mu co-mu terms into typed (intuitionistic) lambda terms via a Continuation-Passing Style transform. The extra expressive power provided by the CPS transform allows lexical items to denote functions that do not correspond to any expression in the lambda mu co-mu language. This in turn allows the lexical items to extend the expressive power of the Lambek/Grishin grammars to simulate the behavior of delimited continuations.

Why would natural language scope-taking require delimited continuations? Well, if scope-taking expressions always took scope at the top level of an utterance, undelimited continuations would be adequate. For instance, in Parigot's lambda mu calculus, which explicitly aims to model classical reasoning, the reduction rules continue to moved named expressions all the way to the top level of the term. That is, in the lambda mu calculus, scope-taking expressions always have maximal (i.e., undelimited) scope. And indeed, this is the way that de Groote's (2001b) analysis of natural language scope-taking works, since he adapts the lambda mu calculus.

But in natural language, it is always possible for scope taking expressions to take scope over a proper subpart of the expression in question.

(1) a.	John saw everyone.	$\forall x.\mathbf{saw}\ x\,\mathbf{j}$
b.	Mary claimed John saw everyone.	$\mathbf{claimed}(\forall x.\mathbf{saw}\ x\,\mathbf{j})(\mathbf{j})$

In (1a), the quantifier introduced by *everyone* takes scope over the entire sentence. But the sentence in (1b) clearly has an interpretation on which the quantifier takes scope only over the embedded clause. That is, (1b) does not say that Mary made as many claims as there are people; rather, Mary made one comprehensive claim, and that claim necessarily involves universal quantification.

The reason embeddability argues that continuations must be delimited is that in order for the result of the embedded computation to be usable in further computations, that result must be typed.

The need for delimited continuations becomes even more clear when we consider cases in which the scope-taking element changes the result type of the expression it takes scope over. This means that no single result type, whether the result type is conceived of as $\perp$ or as $\mathtt{t}$, will suffice. To see what is at issue, consider the following phrase:

(2) a book [the author of which] I know

In this example, the wh-word *which* takes scope over the bracketed determiner phrase. That is, the context *the author of* [ ] has type $\text{DP}\backslash\!\!\backslash\text{DP}$: it is the kind of context that takes a determiner phrase plug and returns a determiner phrase. For instance, if we plug in the determiner phrase *Waverly*, we get the determiner phrase *the author of Waverly*. However, in this case, the bracketed phrase does not function as a determiner phrase (for instance, if we replace the bracketed phrase with a simple determiner phrase, the result is ungrammatical: **a book Waverly I know*). Rather, the bracketed phrase must function as a relative pronoun such as *that* (as in *a book that I know*). Thus if REL is the category of a relative pronoun (whose internal structure need not concern us here; see, e.g., Morrill (1994:xxx) for details), the category of *which* in (2) must be $\text{REL}/\!\!/(\text{DP}\backslash\!\!\backslash\text{DP})$: the sort of expression that plugs into a hole of type DP, takes scope over an DP, and changes it into a REL.

In other words, the use of *which* in (2) changes the expression it takes scope over from a determiner phrase into a relative pronoun. If $\perp$ is our only result type, we are faced with a problem, since we appear to need two distinct result types (namely, DP and REL).

In sum, the fact that scope-taking expressions in natural language are always able to take scope over a proper subpart of the sentences in which they occur, and the fact that scope-taking expressions are able to explicitly change the result type of the expressions they take scope over, strongly suggests that the scope-taking mechanism for natural language must provide delimited continuations. Thus Bernardi and Moortgat (2007) are able to simulate delimited continuations using a system based on undelimited continuations. In our system here, however, we provide delimited continuations directly without needing to extend the expressive power of the logic. Put another way, NL_{CL} (and therefore NL_{λ}) fully characterize delimited scope-taking within the logic itself.

[Show a derivation of (1b) and (2).]

5. Nested contexts and discontinuity

[The remainder of the draft is too rough to share at this point]