NL_λ as the logic of scope and movement

Chris Barker, *New York University*, cb125@nyu.edu

December 21, 2018

Abstract

Lambek elegantly characterized the structure of part of natural language. As is well-known, his substructural logic L, and its non-associative version NL, handle basic composition well ('external merge'), but not scope taking and syntactic displacement ('internal merge'), at least, not in their full generality. In previous work, I propose NL_λ, which is NL supplemented with a single structural inference rule ("abstraction"). Abstraction closely resembles the traditional linguistic rule of Quantifier Raising, and characterizes both semantic scope taking and syntactic displacement. Due to the unconventional form of the abstraction inference, there has been some doubt that NL_λ should count at a legitimate substructural logic. This paper argues that NL_λ is perfectly well-behaved. In particular, it enjoys cut elimination and an interpolation result. In addition, perhaps surprisingly, it is decidable. Finally, I prove that it is sound and complete with respect to the usual class of relational frames.

Keywords: Lambek, substructural logic, scope, syntactic movement, continuations, decidability

Thanks to Glyn Morrill and Greg Restall and my three referees for helpful discussion.

1 NL_λ

NL_λ is a substructural logic (Restall 2000, Ono 2003) with applications in the syntax and semantics of natural languages. It is based on the non-associative Lambek calculus NL (Lambek 1958, 1961; Moortgat 1997, Buszkowski 2010), augmented by a structural postulate ("abstraction") that corresponds to the traditional linguistic rule of Quantifier Raising (May 1977, 1985; Heim and Kratzer 1998).

NL_λ was first proposed in Barker 2007, where it was used to provide a compositional account of the semantics of *same* and *different* in English. The present paper concentrates on the formal properties of NL_λ, rather than on its linguistic applications; for analyses in NL_λ of scope, binding, ellipsis, and more, see chapters 13 through 16 in Barker and Shan 2014, as well as Barker 2007, 2013, 2015a, 2015b, and Barker in prep.

Because the abstraction postulate does not resemble standard structural postulates (for a list of standard structural postulates, see, e.g., Restall 2000:26), there has been some doubt that NL_λ is a legitimate substructural logic. For instance, Kubota 2015:23 remarks that "[t]he use of λs in structured antecedents is without precedent," and he questions the "formal and ontological status" of the postulate. So the present paper sets out to establish that NL$_\lambda$ is perfectly legitimate and well-behaved as a substructural logic.

The mistrust of the abstraction postulate persists despite the fact that Barker and Shan 2014 (section 17.9) prove soundness and completeness and decidability for a restricted version of NL$_\lambda$. However, there are two aspects of that discussion that limit its persuasiveness, both of which are remedied here. One limitation is that their proof is indirect: they first prove that NL$_\lambda$ is equivalent to a more conventional substructural logic (discussed below in section 5.1), and then prove soundness and completeness for the second logic. The proofs here deal directly with NL$_\lambda$. The second limitation is that because their results apply only to a restricted version of NL$_\lambda$, they leave open the status of unrestricted NL$_\lambda$. They remark (page 181) that extending the results to the unrestricted version would require "new techniques". The present paper supplies the new techniques.

The logic has two modes. One mode corresponds to Lambek's original logic, and characterizes horizontal composition, i.e., left/right adjacency (in syntactic terms, "external merge"). The other mode characterizes vertical composition, that is, context/value combination ("internal merge"). The vertical mode allows an expression to combine with one of its delimited continuations, so I will sometimes call this the continuation mode.

The addition of a product unit (as in, e.g., Lambek 1988) for the horizontal mode allows the logic to model gaps left by syntactic movement. Thus abstraction and the unit together allow NL$_\lambda$ to characterize both covert movement (scope) and overt movement (syntactic displacement).

Unrestricted NL$_\lambda$, the logic of scope and movement:

- Formulas: $F := \textsc{dp} \mid \textsc{s} \mid \textsc{q} \mid t \mid F\backslash F \mid F \times F \mid F/F \mid F\backslash\!\!\backslash F \mid F \otimes F \mid F /\!\!/ F$
- Structures: $S := F \mid 1 \mid S \bullet S \mid S \circ S$
- Axiom and cut:

$$\frac{}{A \vdash A}\ \textsc{axiom} \qquad\qquad \frac{\Gamma \vdash A \quad \Sigma[A] \vdash B}{\Sigma[\Gamma] \vdash B}\ \textsc{cut}$$

- Logical inference rules:

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

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

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

$$\frac{\Sigma[1] \vdash A}{\Sigma[t] \vdash A}\ \mathrm{t}L \qquad \frac{}{1 \vdash t}\ \mathrm{t}R$$

- Structural rules:

$$1 \bullet \Delta \equiv \Delta \equiv \Delta \bullet 1 \qquad \text{(unit)}$$
$$\Sigma[\Delta] \equiv \Delta \circ \lambda\alpha\Sigma[\alpha] \qquad \text{(abstraction)}$$

The notation '$\Sigma[\Delta]$' indicates a structure Σ containing within it a distinguished occurrence of a structure Δ, so that '$\Sigma[\Gamma]$' will be the result of replacing the relevant occurrence of Δ in Σ with Γ. The brackets allow replacing only a single occurrence in a context (versus replacing an arbitrary number of occurrences), as usual in discussions of Lambek-style logics (e.g., Moortgat 1997).

In view of the abstraction structural rule, we must enlarge the set of structures by adding the following closure rule: if Δ and $\Sigma[\Delta]$ are structures, and α is a variable from the set $\{x, y, z, \ldots\}$ not occurring in $\Sigma[\,]$, then $\lambda\alpha\Sigma[\alpha]$ is a structure. For instance, $\textsc{dp}$, $\textsc{dp} \bullet \textsc{s}$, $\textsc{dp} \circ \lambda x(x \bullet \textsc{s})$, $\lambda y \lambda x(x \bullet y)$, and $\lambda z(z \bullet \lambda xx)$ are structures, but x, $(x \bullet \textsc{s})$, $\lambda x.\textsc{dp}$, and $\lambda x \lambda x(x \bullet x)$ are not. It will simplify the specification of the logic if we adopt a constraint on derivations inspired by Barendregt's 1984:26 variable convention: we'll assume that no variable occurs in more than one premise. Since we have an unbounded number of distinct variables, this is easy to guarantee. As a result of adopting this constraint, every $\lambda\alpha$ will always correspond to exactly one occurrence of α in the body of the structure. This means that we can safely cut or introduce a slash within the body of a lambda structure without needing to rename any variables.

The abstraction inference rule closely resembles the traditional linguistic rule of Quantifier Raising (May 1977, 1985; Heim and Kratzer 1998). It says that whenever a structure Σ contains within it an occurrence of the structure Δ, Δ can potentially take scope over Σ. This is accomplished here by adjoining Δ in the continuation mode (i.e., via '$\circ$') with the lambda structure $\lambda\alpha\Sigma[\alpha]$, where α is a variable not occurring anywhere in $\Sigma[\,]$. Since the lambda structure $\lambda\alpha\Sigma[\alpha]$ surrounds Δ, it constitutes one of Δ's (delimited) continuations.

As we should expect from the resemblance to Quantifier Raising, NL_λ is well suited to modeling in-situ quantifier scope. If we assign *Ann* to the category $\textsc{dp}$, *saw* to the category $(\textsc{dp}\backslash\textsc{s})/\textsc{dp}$, and the quantifier *everyone* to the category $\textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s})$, we have the following account of in-situ scope taking:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textsc{dp}\vdash\textsc{dp} \qquad \dfrac{\textsc{dp}\vdash\textsc{dp} \quad \textsc{s}\vdash\textsc{s}}{\textsc{dp}\bullet\textsc{dp}\backslash\textsc{s}\vdash\textsc{s}}\,\backslash L}{\textsc{dp}\bullet((\textsc{dp}\backslash\textsc{s})/\textsc{dp}\bullet\textsc{dp})\vdash\textsc{s}}\,/L}{\textsc{dp}\circ\lambda x(\textsc{dp}\bullet((\textsc{dp}\backslash\textsc{s})/\textsc{dp}\bullet x))\vdash\textsc{s}}\,\textsc{abs}}{\lambda x(\textsc{dp}\bullet((\textsc{dp}\backslash\textsc{s})/\textsc{dp}\bullet x))\vdash\textsc{dp}\backslash\!\!\backslash\textsc{s}}\,\backslash\!\!\backslash R \qquad \textsc{s}\vdash\textsc{s}}{\textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s})\circ\lambda x(\textsc{dp}\bullet((\textsc{dp}\backslash\textsc{s})/\textsc{dp}\bullet x))\vdash\textsc{s}}\,/\!\!/L}{\textsc{dp}\bullet((\textsc{dp}\backslash\textsc{s})/\textsc{dp}\bullet\textsc{s}/\!\!/(\textsc{dp}\backslash\!\!\backslash\textsc{s}))\vdash\textsc{s}}\,\textsc{abs}}{\text{Ann}\bullet(\text{saw}\bullet\text{everyone})\vdash\textsc{s}}\,\textsc{lex}$$

The natural Curry-Howard labeling (see, e.g., chapter 13 in Barker and Shan 2014) gives the desired semantics for scope-taking, namely, $\mathbf{everyone}(\lambda x.\mathbf{saw}(x)(\mathbf{ann}))$.

One way to understand the abstraction rule is as a kind of syntactic movement. In this derivation, the scope-taker *everyone* moves upwards to adjoin to its scope domain. A movement analogy is not necessary, however. It is equally legitimate to say that the scope-taker remains in place, and we are simply focussing it, and placing the context that surrounds it into the background. See Barker and Shan 2014, Barker 2015a, 2015b for discussion.

With the addition of a unit for the horizontal mode, the logic also provides an account of overt syntactic movement. The unit structural rule says that the structural constant ‘1’ (which can be thought of as the empty structure, ‘()’) serves as both a left and right unit for the horizontal mode. The unit will be used to model gaps for overt movement as well as silent modifiers, and plays a key role in sprouting in the analysis of sluicing in Barker 2013 and chapter 15 of Barker and Shan 2014. The unit structure corresponds to the type ‘*t*’. The symbol ‘*t*’ stands for the weakest tautology, as in, e.g., Restall 2000, but in a happy coincidence, it also corresponds to the symbol often used in the syntactic literature to stand for syntactic traces.

For example, if we assign *who* to the category $\text{Q}/(t\otimes(\text{DP}\backslash\!\!\backslash\text{S}))$, then following Morrill et al. 2011:22, we can analyze an embedded question such as *who Ann saw*, as in *Bill knows [who Ann saw]* as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{1\vdash t \qquad \begin{array}{c}\vdots\\ \lambda x(\text{Ann}\bullet(\text{saw}\bullet x))\vdash \text{DP}\backslash\!\!\backslash\text{S}\end{array}}{1\circ\lambda x(\text{Ann}\bullet(\text{saw}\bullet x))\vdash t\otimes(\text{DP}\backslash\!\!\backslash\text{S})}\otimes R}{\text{Ann}\bullet(\text{saw}\bullet 1)\vdash t\otimes(\text{DP}\backslash\!\!\backslash\text{S})}\text{ABS}}{\text{Ann}\bullet\text{saw}\vdash t\otimes(\text{DP}\backslash\!\!\backslash\text{S})}\text{unit} \qquad \text{Q}\vdash\text{Q}}{\text{Q}/(t\otimes(\text{DP}\backslash\!\!\backslash\text{S}))\bullet(\text{Ann}\bullet\text{saw})\vdash\text{Q}}\backslash L$$

In the traditional syntactic treatment, the wh-word *who* moves from the object position of *saw*, leaving a trace. That movement corresponds here to the abstraction of the unit, establishing a logical connection between the object position and the wh-word. The *t* in the lexical category of *who* licenses the abstraction of the unit.

As mentioned above, Barker and Shan 2014 place restrictions on the abstraction rule that are not imposed here (see their section 17.6, page 190). Essentially, they allow abstraction only out of contexts built from the horizontal mode, but not out of contexts built from the vertical mode. This restriction allows Barker and Shan to prove an Efficient Abstraction theorem, which says that for every derivation, there is an equivalent derivation in which only formulas (as opposed to complex structures) undergo abstraction (i.e., Δ in the abstraction rule is always a formula). As they note, this has the unfortunate effect of ruling out derivations proposed for sluicing in Barker 2013 (see discussion in Barker and Shan 2014:205, section 7.11), since sluicing crucially requires abstraction of a lambda structure. The unrestricted version of NL$_\lambda$ given here covers all of the sluicing analyses proposed in Barker 2013, and the decidability proof below does not depend on Efficient Abstraction.

The presence of lambdas and variables in the syntax of NL$_\lambda$ often creates concern; for an example in print, once again, see Kubota 2015:23. Are these lambdas “real” lambdas? And what about the variables? Although these formal elements are novel in the context of the syntactic side of Lambek grammars, variables and binding constructions are common in logic in general (just think of the Predicate Calculus with quantification). Furthermore, there is a related literature that uses lambda expressions in syntactic calculi. Notable examples include Oehrle 1994, Muskens 2001, and de Groote 2002. In any case, the set of formulas in the logic are well-defined, the set of structures in the logic are well-defined, and the inferences licensed by the abstraction postulate are well-defined. Just like any structural rule, the abstraction rule licenses replacing one structure with a different structure based on purely structural criteria.

In other words, just like any logic, NL_λ fully and precisely characterizes a class of inferences. And just as we can for any logic, we can ask the usual metalogical questions: does the logic enjoy cut elimination? (Yes.) Is the logic decidable? (Surprisingly, yes.) Is there an interpolation theorem? (Yes.) What sort of semantics fits the logic? (The same relational frames that are often used to provide a semantics for other substructural logics.) Is the logic sound and complete with respect to its semantics? (Yes.)

2 Decidability

Note that NL_λ has the subformula property. This is easy to see, since NL has the subformula property, and the structural postulates neither add nor delete any formulas. The subformula property often leads to decidability, as long as there is a bound on the application of the structural postulates. For instance, there are only so many ways of regrouping a set of structures, so associativity does not threaten decidability for L.

In contrast with L, it is not obvious that NL_λ is decidable, since abstraction inferences can be repeated ad infinitum (e.g., $A \equiv A \circ \lambda xx \equiv (A \circ \lambda xx) \circ \lambda xx \equiv ...$). Without some way of limiting abstraction, proof search will continue forever. Proving decidability will involve three main lemmas: cut elimination, a bound on the number of abstractions, and a bound on the number of units.

2.1 Cut elimination lemma

Lemma 1 (Cut elimination): Every provable sequent has a cut-free proof.

Proof. The standard proof of cut admissibility for NL relies on pushing cuts upward in the proof until one of the premises is an axiom, at which point both the axiom and the cut can be eliminated without disturbing the proof. The logical rules of NL_λ are the same as the rules for other Lambek-style grammars, and so are equally compatible with cut elimination. As for the abstraction postulate, since every formula present in the conclusion of an abstraction is also present in its premise, any cut that targets a formula in the conclusion can be pushed upwards to target the corresponding formula in the premise, so abstraction inferences do not impede cut elimination. □

See Barker and Shan 2014 chapter 17 for a more detailed version of this proof.

Note the material that replaces the cut formula in the conclusion of a cut inference must constitute the entire left hand side of one of the premises, so there is no danger of a cut separating a variable from its binder.

2.2 Abstraction lemma

Since the abstraction inference rule is bi-directional, we can separate it into two parts, corresponding to beta reduction (‘RED’) and beta expansion (‘EXP’):

$$\frac{\Sigma[\Gamma[\Delta]] \vdash A}{\Sigma[\Delta \circ \lambda x \Gamma[x]] \vdash A}\,\text{RED} \qquad \frac{\Sigma[\Delta \circ \lambda x \Gamma[x]] \vdash A}{\Sigma[\Gamma[\Delta]] \vdash A}\,\text{EXP}$$

The terms ‘reduction’ and ‘expansion’ reflect the bottom-to-top direction of fit, since decidability is a matter of proof search, which involves starting with the final conclusion and searching to find a proof that licenses it.

Since the premise of a reduction inference is strictly simpler than its conclusion, it poses no threat to decidability. The goal, then, will be to establish a bound on the number of expansions needed.

The strategy for proving the lemma stated immediately below will depend on tracking the lifetime within the proof of each ‘$\circ$’. Since there are no ‘$\circ$’s in axiom instances, each ‘$\circ$’ must be introduced into the derivation either by a logical rule (more specifically, by $\backslash\!\!\backslash L$, $\otimes R$, or $/\!\!/L$) or by a reduction inference. Once a ‘$\circ$’ has been introduced, it will be copied by subsequent inference rules from premise to conclusion until it is eliminated either by a logical rule ($\backslash\!\!\backslash R$, $\otimes L$, or $/\!\!/R$), or by an expansion inference. I’ll say that a chain of $\circ$’s that is introduced by a reduction and eliminated by an expansion is a *purely structural* abstraction.

Lemma 2 (purely structural abstraction is eliminable): given an arbitrary derivation in NL$_\lambda$, there is an equivalent derivation (same conclusion, same Curry-Howard labeling) in which no ‘$\circ$’ eliminated by an expansion inference was introduced by a reduction inference.

Proof. In view of Lemma 1, we can assume without any loss of generality that we have a cut-free proof. If the derivation does not contain any purely structural abstractions, the derivation is already consistent with the lemma. Therefore assume that there is at least one purely structural abstraction in the proof. We will construct an equivalent parallel proof in which the purely structural abstraction is omitted.

$$\begin{array}{cc}
\begin{array}{c}
\vdots \\
\dfrac{\Sigma[\Gamma[\Delta]] \vdash A}{\Sigma[\Delta \circ \lambda x \Gamma[x]] \vdash A}\,\text{RED} \\
\vdots \\
\dfrac{\Sigma'[\Delta' \circ \Pi'[\lambda x \Gamma'[x]]] \vdash A'}{\Sigma''[\Delta'' \circ \Pi''[\lambda x \Gamma''[x]]] \vdash A''}\,\text{X} \\
\vdots \\
\dfrac{\Sigma'''[\Delta''' \circ \lambda x \Gamma'''[x]] \vdash A'''}{\Sigma'''[\Gamma'''[\Delta''']] \vdash A'''}\,\text{EXP} \\
\vdots
\end{array}
&
\begin{array}{c}
\vdots \\
\Sigma[\Gamma[\Delta]] \vdash A \\
\vdots \\
\dfrac{\Sigma'[\Pi'[\Gamma'[\Delta']]] \vdash A'}{\Sigma''[\Pi''[\Gamma''[\Delta'']]] \vdash A''}\,\text{X} \\
\vdots \\
\Sigma'''[\Gamma'''[\Delta''']] \vdash A''' \\
\vdots
\end{array}
\end{array}$$

In the original proof on the left, a reduction introduces a ‘$\circ$’ that is carried through until a matching expansion eliminates it. The claim is that it is always possible to construct a corresponding modified proof as schematized on the right with identical starting and ending sequents, but without the abstraction.

Examining the reduction inference in the original derivation, note that in the conclusion of the inference, the '$\circ$' is adjacent to the newly created λ. Subsequent inferences can interpose structure between a '$\circ$' and its lambda. For instance, additional reductions can abstract material from Γ into this position, and we can insert as many units as we wish. The possibility of interpolated material is schematically represented in the middle inference as Π' and Π''. However, in order for the final expansion to eliminate the '$\circ$', the '$\circ$' must be reunited with its original λ, since inspection of the logic reveals there is no way to create or maneuver any different lambda into the position adjacent to a pre-existing '$\circ$'. This means in addition that if the structure $\lambda x\Gamma[x]$ created by the orginal reduction is itself abstracted, it will only have to undergo expansion back into its original position in order for the occurrence of $\circ$ in question to be eliminated, so this secondary abstraction would itself be removable.

Focus now on the schematic inference labelled X. The intention is that X can be any inference, either logical or structural (and so may have a second premise, not represented in the schematic diagram). Assuming that the premise in the original proof and the premise in the constructed proof share Σ', Π', Γ', and Δ', the premises differ only in the position of Δ', the presence of the $\circ$ and the λx, and the occurrence of x embedded in Γ'. In particular, every formula, every structural connective, and every potential landing site for an abstraction that is present in the left side premise is present also in the right side premise. It follows that the right hand premise will match the requirements of the X inference, and that the left hand conclusion and the right hand conclusion will also share identical Σ'', Π'', Γ'', and Δ'' (up to choice of variables).

In particular, it is not possible for $\otimes L$ and $/\!\!/R$ inferences to target the distinguished '$\circ$', since the right hand element is not a formula. A $\backslash\!\!\backslash R$ inference can potentially target the '$\circ$'—but then the chain containing the '$\circ$' would not be purely structural, contrary to assumption.

Since the subproofs begin synchronized (by construction), and since each step maintains synchronization, the final conclusions will be identical. Furthermore, because the same logical inferences are executed in the same way (that is, they manipulate the same formulas to build the same structures), and in the same order, and because structural inferences do not affect the Curry-Howard labeling, the semantic label of the final sequent of the original proof will be identical to that of the constructed proof. $\square$

Given lemma 2, it follows that we can limit proof search to derivations in which each '$\circ$' eliminated by an expansion inference was introduced by a logical rule. Because each logical rule introduces at most one '$\circ$', and because the number of instances of the logical rules is bounded by the number of logical connectives in the final sequent, it follows that the number of expansion inferences is also bounded by the number of logical connectives in the final sequent.

2.3 Unit lemma

Just as we did for abstraction, we can separate the structural rule governing the unit into operations that introduce ("push") and eliminate ("pop") a unit:

$$\frac{\Sigma[\Delta] \vdash A}{\Sigma[1 \bullet \Delta] \vdash A}\ \textsc{left push} \qquad \frac{\Sigma[\Delta] \vdash A}{\Sigma[\Delta \bullet 1] \vdash A}\ \textsc{right push}$$

$$\frac{\Sigma[1 \bullet \Delta] \vdash A}{\Sigma[\Delta] \vdash A}\ \textsc{left pop} \qquad \frac{\Sigma[\Delta \bullet 1] \vdash A}{\Sigma[\Delta] \vdash A}\ \textsc{right pop}$$

The inference names are from Restall 2000:30; the choice of '1' for the unit comes from Ono 2003.

Only inferences in which the premises are more complex than the conclusion (i.e., the pop rules) threaten decidability, so we can concentrate on them.

Lemma 3 (a bounded number of pops suffices): given an arbitrary derivation in NL$_\lambda$, there is an equivalent derivation (same conclusion, same Curry-Howard labeling) in which the number of pop inferences is bounded by the length of the final conclusion.

Proof. Let p be an arbitrary derivation. In view of cut admissibility, we can assume that p is cut-free. Focus on an instance of pop, and on the inference i immediately above it. In most instances, the order of pop and i can be exchanged without affecting the proof (same final conclusion, same Curry-Howard labeling). But there are a limited number of situations in which the exchange cannot occur. Here is a complete list:

- Pop blocked by an $\backslash L$, $\times R$, or $/L$ inference:

$$\frac{\dfrac{1 \vdash A \quad \Sigma[B] \vdash C}{\Sigma[1 \bullet A\backslash B] \vdash C}\ \backslash L}{\Sigma[A\backslash B] \vdash C}\ \textsc{left pop} \qquad \frac{\dfrac{1 \vdash A \quad \Sigma[B] \vdash C}{\Sigma[B/A \bullet 1] \vdash C}\ /L}{\Sigma[B/A] \vdash C}\ \textsc{right pop}$$

$$\frac{\dfrac{1 \vdash A \quad \Delta \vdash B}{1 \bullet \Delta \vdash A \times B}\ \times R}{\Delta \vdash A \times B}\ \textsc{left pop} \qquad \frac{\dfrac{\Delta \vdash A \quad 1 \vdash B}{\Delta \bullet 1 \vdash A \times B}\ \times R}{\Delta \vdash A \times B}\ \textsc{right pop}$$

- Pop blocked by an expansion inference (four subcases):

$$\frac{\dfrac{\Sigma[1 \circ \lambda x \Gamma[x \bullet \Delta]] \vdash A}{\Sigma[\Gamma[1 \bullet \Delta]] \vdash A}\ \textsc{exp}}{\Sigma[\Gamma[\Delta]] \vdash A}\ \textsc{left pop} \qquad \frac{\dfrac{\Sigma[1 \circ \lambda x \Gamma[\Delta \bullet x]] \vdash A}{\Sigma[\Gamma[\Delta \bullet 1]] \vdash A}\ \textsc{exp}}{\Sigma[\Gamma[\Delta]] \vdash A}\ \textsc{right pop}$$

$$\frac{\dfrac{\Sigma[\Delta \circ \lambda x \Gamma[1 \bullet x]] \vdash A}{\Sigma[\Gamma[1 \bullet \Delta]] \vdash A}\ \textsc{exp}}{\Sigma[\Gamma[\Delta]] \vdash A}\ \textsc{left pop} \qquad \frac{\dfrac{\Sigma[\Delta \circ \lambda x \Gamma[x \bullet 1]] \vdash A}{\Sigma[\Gamma[\Delta \bullet 1]] \vdash A}\ \textsc{exp}}{\Sigma[\Gamma[\Delta]] \vdash A}\ \textsc{right pop}$$

The top two cases, in which it is the unit that is abstracted, correspond to using the unit as a trace for syntactic movement, as illustrated in section 1. The bottom two cases, in which it is the complement of the unit that is abstracted, correspond to the analysis of sprouting in sluicing (see Barker 2013:section 4.3 or Barker and Shan 2014:182 for an analysis of the sprouting sluice in *John left, but I don't know when*; the unit serves as a silent temporal adverbial modifying *left*).

Note that these cases can overlap if $\Delta = 1$:

$$\dfrac{\dfrac{\dfrac{1\circ\lambda x\Gamma[(x\bullet 1)\bullet\Pi]\vdash A}{\Gamma[(1\bullet 1)\bullet\Pi]\vdash A}\,\text{EXP}}{\Gamma[1\bullet\Pi]\vdash A}\,\text{LEFT POP}_2}{\Gamma[\Pi]\vdash A}\,\text{LEFT POP}_1 \qquad \dfrac{\dfrac{\dfrac{1\circ\lambda x\Gamma[(1\bullet x)\bullet\Pi]\vdash A}{\Gamma[(1\bullet 1)\bullet\Pi]\vdash A}\,\text{EXP}}{\Gamma[1\bullet\Pi]\vdash A}\,\text{LEFT POP}_2}{\Gamma[\Pi]\vdash A}\,\text{LEFT POP}_1$$

The left-hand (right-hand) derivation matches both of the schemas in which the variable x occupies the leftmost (rightmost) position. In this one kind of situation, a single expansion inference can simultaneously block two pop inferences. We can exchange the order of the pop inferences with each other in order to select which inference is adjacent to the expansion inference; but neither one can hop over the expansion inference, so both pop inferences remain trapped beneath the expansion. In all other situations, an inference can block at most one pop inference. There are variants with Π on the left hand side of the units.

- Pop blocked by push (two subcases):

$$\dfrac{\dfrac{\Sigma[\Delta]\vdash A}{\Sigma[1\bullet\Delta]\vdash A}\,\text{LEFT PUSH}}{\Sigma[\Delta]\vdash A}\,\text{LEFT POP} \qquad \dfrac{\dfrac{\Sigma[\Delta]\vdash A}{\Sigma[\Delta\bullet 1]\vdash A}\,\text{RIGHT PUSH}}{\Sigma[\Delta]\vdash A}\,\text{RIGHT POP}$$

Since these proof fragments begin and end with the same sequent (with the same Curry-Howard label), they can be deleted without affecting the proof.

A pop inference is never blocked by another pop inference, though one case needs comment:

$$\dfrac{\dfrac{\Sigma[(1_1\bullet 1_2)\bullet\Delta]\vdash A}{\Sigma[1_1\bullet\Delta]\vdash A}\,\text{RIGHT POP}_2}{\Sigma[\Delta]\vdash A}\,\text{LEFT POP}_1 \not\equiv \dfrac{\dfrac{\Sigma[\Delta\bullet(1_1\bullet 1_2)]\vdash A}{\Sigma[\Delta\bullet 1_2]\vdash A}\,\text{LEFT POP}_1}{\Sigma[\Delta]\vdash A}\,\text{RIGHT POP}_2$$

The order of the left pop and right pop inferences can't be exchanged. However, the problematic pair of inferences is equivalent to a pair that clearly can be safely swapped:

$$\dfrac{\dfrac{\Sigma[(1_1\bullet 1_2)\bullet\Delta]\vdash A}{\Sigma[1_1\bullet\Delta]\vdash A}\,\text{RIGHT POP}_2}{\Sigma[\Delta]\vdash A}\,\text{LEFT POP}_1 \equiv \dfrac{\dfrac{\Sigma[(1_2\bullet 1_1)\bullet\Delta]\vdash A}{\Sigma[1_1\bullet\Delta]\vdash A}\,\text{LEFT POP}_2}{\Sigma[\Delta]\vdash A}\,\text{LEFT POP}_1$$

Once we replace the left hand proof fragment with the equivalent fragment on the right, we can safely swap the Left Pop$_2$ inference with the Left Pop$_1$ inference. By the same kind of reasoning, a right push can cancel with a left pop when the target has the form $\Sigma[1\bullet 1]$.

With these observations in hand, we can push pop inferences as far upwards in the proof as possible. If a pop is blocked, it will get trapped immediately beneath the inference that blocks it. If the blocking inference is a push inference, the two inferences cancel out, and we can remove them. Since there are a bounded number of $\backslash L$, $/L$, and expansion inferences, there will be a bounded number of pops. $\square$

Unlike abstraction, there are situations in which a purely structural unit—that is, a unit that is both introduced and eliminated via structural rule—cannot be eliminated. The analysis of sprouting in Barker 2013 (reproduced in Barker and Shan 2014, chapter 16) is an example. In that analysis, the introduction of a unit serves the role of a silent modifier that allows remnant movement to provide an antecedent for the sluice gap.

2.4 A decision algorithm

In order to complete the proof of decidability, it is necessary to exhibit an algorithm that is guaranteed to either find a derivation if one exists—or else to guarantee that no derivation is possible—in an amount of time bounded by the size of the sequent to be proved.

Theorem 1 (decidability): NL_λ is decidable.

Proof. By lemma 1, we can limit proof search to cut-free proofs. By lemma 2, we can associate each sequent s with a budget k of expansion inferences, where k is the number of '$\backslash\!\!\backslash$'s, '$/\!\!/$'s, and '$\otimes$'s in s. We can then conduct the normal Gentzen-style proof search, positing expansions freely at every stage, as long as the number of expansions does not exceed the budget for the subproof in question. Since there are a finite number of ways to apply the expansion inference, this search is guaranteed to terminate. By lemma 3, similar reasoning applies to pop inferences for the unit, except that the budget for pop inferences is bounded above by $(2*k)$ + the number of instances of $\backslash L$, $\times R$, and $/L$ in s. $\square$

Although simple, this algorithm is crude. Many additional optimizations are possible. However, a detailed analysis of efficient parsing for NL_λ will have to wait for a different occasion.

3 Polymorphism and an interpolation theorem

Interpolation has a long history in logic, beginning with Craig's 1957 interpolation lemma. In the study of Lambek systems, interpolation has been used to prove bounds on expressivity or computational complexity (e.g., Pentus 1993, building on Roorda 1991; Jäger 2005; Buszkowski 2005). I will use an interpolation result here to understand how lambda structures function in the context of a proof. The result will also be used in the completeness proof below.

In many Lambek systems, there is a simple correspondence between structures and formulas (see, e.g., Buszkowski 2010:13): just map $\bullet$ to $\times$, $\circ$ to $\otimes$, and 1 to t, and anything that can be proven by a structure can be proven by replacing that structure with its corresponding formula, and vice versa.

Extending this strategy to NL_λ is not simple. The reason is that there is no single formula that can behave in all circumstances like a given lambda structure. Lambda structures connect the type of their sibling argument with the type of the containing structure, without fixing in advance what the transmitted type is. For instance, $A \circ \lambda xx \vdash A$ for all choices of A. It follows that $\lambda xx \vdash$

$A\backslash\!\!\backslash A$, so all formulas of the form $A\backslash\!\!\backslash A$ have an equal claim to represent part of the provability potential of λxx. And in fact, this polymorphism is an essential part of how the logic does useful linguistic work, since the structure that undergoes abstraction in a reduction inference when a lambda is introduced can differ arbitrarily from the structure that undergoes expansion when that same lambda is eliminated.

However, when a lambda structure is embedded in a specific proof, we *can* find a formula that accurately tracks the role the structure plays in that proof. It might be helpful to think of a lambda structure not as a type, but as a type operator, that is, as a function from types to types (as in logics that inhabit the rear plane of the Barendregt cube).

Theorem 2 (interpolation): Given $\Sigma[\Delta] \vdash A$, there is a formula δ such that $\Delta \vdash \delta$ and $\Sigma[\delta] \vdash A$.

Proof. The proof proceeds by induction on the number of inferences in the derivation of $\Sigma[\Delta] \vdash A$. Clearly the theorem holds for axiom instances. Assume that the theorem holds for derivations containing $n-1$ inferences. Then we can assign formulas to structural elements in the conclusion of an arbitrary n^{th} inference along the following lines:

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

$$\frac{\Sigma[\gamma{:}\Gamma[\pi{:}\Pi[\delta{:}\Delta]]] \vdash A}{\Sigma[\delta{:}\Delta \circ \delta\backslash\!\!\backslash\gamma{:}\lambda x\Gamma[\delta\backslash\!\!\backslash\pi{:}\Pi[x]]] \vdash A}\,\text{RED} \qquad \frac{\Sigma[\omega{:}\Omega \circ \gamma{:}\lambda x\Gamma[\pi{:}\Pi[x]]] \vdash A}{\Sigma[\omega\otimes\gamma{:}\Gamma[\omega\otimes\pi{:}\Pi[\omega{:}\Omega]]] \vdash A}\,\text{EXP}$$

For structures of the form $\gamma{:}\lambda x\Gamma[\pi{:}\Pi[x]]$, the goal is to find a label π for the incomplete structure $\Pi[x]$ such that $\lambda x\Gamma[x \circ \pi] \vdash \gamma$. For instance, in the labeling for the reduction inference, $\lambda x\Gamma[x \circ \delta\backslash\!\!\backslash\pi]] \vdash \delta\backslash\!\!\backslash\gamma$, as desired. For expansion, we reason as follows:

$$\dfrac{\dfrac{\dfrac{\omega \vdash \omega \qquad \lambda x\Gamma[x \circ \pi] \vdash \gamma}{\omega \circ \lambda x\Gamma[x \circ \pi] \vdash \omega \otimes \gamma}\,\otimes R}{\Gamma[\omega \circ \pi] \vdash \omega \otimes \gamma}\,\text{EXP}}{\Gamma[\omega \otimes \pi] \vdash \omega \otimes \gamma}\,\otimes L$$

The second premise in the top line follows from the inductive hypothesis. □

4 A sound and complete semantics for NL$_\lambda$

NL$_\lambda$ is sound and complete with respect to the usual class of relational frames.

A *frame* $\mathscr{F}$ for NL$_\lambda$ consists of a set of points P, a partial order '$\leq$' on P, and two 3-place accessibility relations, $R^\bullet$ and R°, one for each mode. The accessibility relations can be any three-place relation over P, as long as they satisfy the following requirement for their interaction with the partial ordering $\leq$: for all $abcdef$ with $d \leq a$, $e \leq b$ and $c \leq f$, $R^\bullet abc \to R^\bullet def$, and likewise for R°. That is, the accessibility relations must be downward monotonic with respect to each of their first two arguments, and upward monotonic with respect to their third argument. This is what Restall 2000:240 calls a 'plump' accessibility relation.

A *model* $\mathscr{M}$ for NL$_\lambda$ is a frame plus an evaluation relation $\Vdash$ that relates points to formulas and to structures. For atomic formulas, the only restriction on the evaluation relation is that it is upward monotonic with respect to the partial order $\leq$. That is, if $p \Vdash A$ for any atomic formula A, and $p \leq q$, then $q \Vdash A$. In addition, the evaluation relation must obey the following semantic constraints on complex formulas and on structures:

$$\begin{array}{l}
a \Vdash C/B \text{ iff } \forall bc.(R^\bullet abc \wedge b \Vdash B) \to c \Vdash C \\
b \Vdash A\backslash C \text{ iff } \forall ac.(R^\bullet abc \wedge a \Vdash A) \to c \Vdash C \\
c \Vdash A \times B \text{ iff } \exists ab.R^\bullet abc \wedge a \Vdash A \wedge b \Vdash B \\
a \Vdash C /\!\!/ B \text{ iff } \forall bc.(R^\circ abc \wedge b \Vdash B) \to c \Vdash C \\
b \Vdash A \backslash\!\!\backslash C \text{ iff } \forall ac.(R^\circ abc \wedge a \Vdash A) \to c \Vdash C \\
c \Vdash A \otimes B \text{ iff } \exists ab.R^\circ abc \wedge a \Vdash A \wedge b \Vdash B \\
c \Vdash \Sigma \bullet \Delta \text{ iff } \exists ab.R^\bullet abc \wedge a \Vdash \Sigma \wedge b \Vdash \Delta \\
c \Vdash \Sigma \circ \Delta \text{ iff } \exists ab.R^\circ abc \wedge a \Vdash \Sigma \wedge b \Vdash \Delta \\
i \Vdash 1 \text{ iff } i \Vdash t
\end{array}$$

$$\begin{array}{l}
c \Vdash \Sigma[\Delta] \text{ iff } \exists ab.R^\circ abc \wedge a \Vdash \Delta \wedge b \Vdash \lambda x \Sigma[x] \\
b \Vdash \lambda x \Sigma[x] \text{ only if } \forall ac\Delta.(R^\circ abc \wedge a \Vdash \Delta) \to c \Vdash \Sigma[\Delta]
\end{array}$$

The clauses for the logical connectives and the structural connectives and the unit are standard (see, e.g., Moortgat 1997 or Restall 2000).

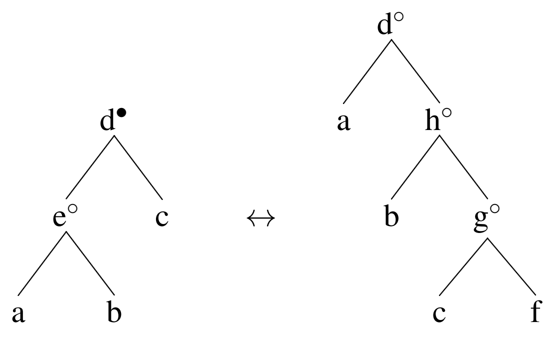
The final two clauses give the semantics for lambda structures. The first of the two resembles the clause for '$\otimes$' structures, and the second is essentially a polymorphic generalization of the clause for '$\backslash\!\!\backslash$' structures. The reason that the second clause is 'only if' instead of 'iff' is that some lambda structures match both clauses, and satisfying the second clause is not enough to satisfy the first clause. One simple option for making the definitions necessary and sufficient is to assume that $\Vdash$ is the largest relation consistent with the constraints.

In order for the semantics to accurately model the structural postulates, it is necessary to constrain the accessibility relations $R^\bullet$ and R° in every frame as follows:

$$\begin{array}{rr}
\forall ab.a \leq b \leftrightarrow (\exists i.R^\bullet iab \wedge i \Vdash t) & \text{(left unit)} \\
\forall ab.a \leq b \leftrightarrow (\exists i.R^\bullet aib \wedge i \Vdash t) & \text{(right u't)} \\
\forall ab.a \leq b \leftrightarrow (\exists i.R^\circ aib \wedge i \Vdash \lambda xx) & \text{(I)} \\
\forall abcd.(\exists e.R^\circ abe \wedge R^\bullet ecd) \leftrightarrow (\exists fgh.R^\circ cfg \wedge R^\circ bgh \wedge R^\circ ahd \wedge f \Vdash \lambda zyx((x \circ y) \bullet z)) & (\text{B}^\bullet) \\
\forall abcd.(\exists e.R^\circ abe \wedge R^\circ ecd) \leftrightarrow (\exists fgh.R^\circ cfg \wedge R^\circ bgh \wedge R^\circ ahd \wedge f \Vdash \lambda zyx((x \circ y) \circ z)) & (\text{B}^\circ) \\
\forall abcd.(\exists e.R^\circ abe \wedge R^\bullet ced) \leftrightarrow (\exists fgh.R^\circ cfg \wedge R^\circ bgh \wedge R^\circ ahd \wedge f \Vdash \lambda zyx(z \bullet (x \circ y))) & (\text{C}^\bullet) \\
\forall abcd.(\exists e.R^\circ abe \wedge R^\circ ced) \leftrightarrow (\exists fgh.R^\circ cfg \wedge R^\circ bgh \wedge R^\circ ahd \wedge f \Vdash \lambda zyx(z \circ (x \circ y))) & (\text{C}^\circ)
\end{array}$$

The names I, B, and C are from combinatory logic (see section 5.1 below).

A diagram will help organize the details of, for example, frame condition B$^\bullet$:



Frame condition B$^\bullet$ guarantees that whenever there are points a,b,c,d,e related by $R^\bullet$ and R° as on the left, there will be points a,b,c,d,f,g,h related as on the right, and vice versa, where f supports the combinator that encodes the correspondence.

4.1 Soundness

A sequent $X \vdash A$ is *valid* if $x \Vdash X \rightarrow x \Vdash A$ in every frame. A logic is *sound* with respect to a class of frames just in case every provable sequent is valid.

In order to prove that NL$_\lambda$ is sound with respect to the semantics and the frame conditions given above, we must first prove that the semantics guarantees monotonicity (cf. Restall 2000:242).

Lemma 4 (monotonicity): If $p \Vdash \Gamma$ and $p \leq q$, then $q \Vdash \Gamma$.

Proof. The lemma holds for atomic formulas by stipulation. If $\Gamma = \Sigma \bullet \Delta$, there exist a and b such that $a \Vdash \Sigma$, $b \Vdash \Delta$, and $R^\bullet abp$. But since $R^\bullet$ is plump, it is upward monotonic in its third argument, so it follows that $R^\bullet abq$. Similarly for structures built from $\circ$, $\times$, or $\otimes$. If $\Gamma = A \backslash C$, then $p \Vdash \Gamma$ iff for all a and c, $(R^\bullet apc \wedge a \Vdash A) \rightarrow c \Vdash C$. But since $R^\bullet aqc \rightarrow R^\bullet apc$, it follows that $p \Vdash \Gamma$ only if $q \Vdash \Gamma$. Similarly for structures built from $/$, $\backslash\!\!\backslash$, and $/\!\!/$. If $\Gamma = \lambda x \Sigma[x]$, there are two possibilities. If there is some structure Δ within Γ, then $p \Vdash \Gamma[\Delta]$ iff $p \Vdash \Delta \circ \lambda y \Gamma[y]$, and the argument for $\circ$ applies. If there is no such Δ, then $p \Vdash \Gamma$ iff for all a, c, and Δ, $R^\circ apc$ and $a \Vdash \Delta$ guarantees $c \Vdash \Sigma[\Delta]$. But again, since $R^\circ aqc \rightarrow R^\circ apc$, it follows that $p \Vdash \Gamma$ only if $q \Vdash \Gamma$. $\square$

Theorem 3 (soundness): NL$_\lambda$ is sound.

Proof. The proof proceeds by induction on proof length. Axiom instances are trivially sound, and it is easy to show for each logical inference rules that whenever the premises are sound, the conclusion is also sound. We only need to worry here about the structural inferences.

For the left unit, we must show that $q \Vdash \Delta$ iff $q \Vdash 1 \bullet \Delta$. In the left to right direction, assume that $q \Vdash \Delta$. Then $q \Vdash 1 \bullet \Delta$ iff there exists i and p such that $i \Vdash 1$, $p \Vdash \Delta$, and $R^\bullet ipq$. According to the evaluation condition for the unit, $i \Vdash 1$ iff $i \Vdash t$. Frame condition (left unit) guarantees the existence of a point i such that $i \Vdash t$ and $R^\bullet ipq$ iff $p \leq q$. If we choose $p = q$, we have $i \Vdash 1$ and $R^\bullet ipq$, so $q \Vdash 1 \bullet \Delta$.

In the right to left direction, assume $q \Vdash 1 \bullet \Delta$. Then there are i and p such that $i \Vdash t$, $p \Vdash \Delta$, and $R^\bullet ipq$. But by the frame condition, the existence of such an i entails $p \leq q$. This is enough to guarantee that $q \Vdash \Delta$, in view of lemma 4.

A similar argument holds for the right unit.

For abstraction, we need to show that for all points c and all structures Δ and $\Sigma[\Delta]$, $c \Vdash \Sigma[\Delta]$ iff there exist points a and b such that $a \Vdash \Delta$, $b \Vdash \lambda x \Sigma[x]$, and $R^\circ abc$. The right to left direction (i.e., expansion) follows immediately from the semantic clauses.

In order to prove the left to right direction (reduction), say that a structure is *open* just in case no substructure is contained within a lambda structure. We can first establish the soundness of reduction for open $\Sigma[\Delta]$, and then extend that result to the general case. The proof proceeds by induction on the structural complexity of $\lambda x\Sigma[x]$. For the base case, $\lambda x\Sigma[x] = \lambda xx$, and frame condition I guarantees the existence of suitable points. If $\Sigma[\Delta] = \Pi[\Delta] \bullet \Gamma$ (similar reasoning applies to the other three possible cases), then we have

$$\begin{array}{lr} c \Vdash \Pi[\Delta] \bullet \Gamma & \\ \text{iff } c \Vdash (\Delta \circ \langle \lambda x\Pi[x] \rangle) \bullet \Gamma & \text{(inductive hypothesis)} \\ \text{iff } c \Vdash \Delta \circ (\langle \lambda x\Pi[x] \rangle \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z))) & \text{(frame condition B}^\bullet\text{)} \end{array}$$

Here, '$\langle \lambda x\Pi[x] \rangle$' is a structure that is guaranteed to be semantically equivalent to $\lambda x\Pi[x]$. Note that the induction guarantees that if the starting point is open, there will be a structure $\langle \lambda x\Pi[x] \rangle$ that is open, in which case $\langle \lambda x\Pi[x] \rangle \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z))$ will also be open.

We need to show that for all points b, $b \Vdash (\langle \lambda x\Pi[x] \rangle \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z)))$ iff $b \Vdash \lambda x(\Pi[x] \bullet \Gamma)$. In order to do this, we must consider the two relevant semantic clauses. Since the choice of Δ was arbitrary, the only-if semantic clause is satisfied. As for the other semantic clause, we need to show that for all open Δ, $b \Vdash (\langle \lambda x\Pi[x] \rangle \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z)))[\Delta]$ iff $b \Vdash \Delta \circ \lambda y(\lambda x\Pi[x] \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z)))[y]$. Since $(\langle \lambda x\Pi[x] \rangle \circ (\Gamma \circ \lambda zyx((x \circ y) \bullet z)))[\Delta]$ is open, the result follows from reapplication of the reasoning just given.

In order to extend the result to non-open structures, consider a proof whose soundness we wish to guarantee. We can construct a parallel proof in which each structure is open. Here's how: at the stage of the proof at which the first reduction inference occurs, all structures are still open. The reasoning just given shows how to construct an open structure that is guaranteed to be semantically equivalent to the the lambda structure created by the reduction. Each structure in the original proof will correspond to a semantically equivalent open structure in the parallel proof. As a result, every subsequent inference that targets structures in the original proof will apply equally well to the corresponding elements in the parallel proof, with semantically identical results. When the proof arrives at subsequent reduction inferences, we repeat the process in the parallel proof, and so on. Because the parallel proof with open structures guarantees the existence of suitable points, the semantic equivalence with the original proof suffices to establish the soundness of reduction. □

The strategy of mapping lambda structures onto open structures is essentially the standard embedding of lambda terms into combinatory logic (e.g., Barendregt 1984:152); see section 5.1 below.

4.2 Completeness

A logic is (weakly) complete with respect to a class of frames just in case every valid sequent is provable. I will establish completeness in the usual way, by constructing a 'canonical' frame and model that only validates provable sequents.

In some completeness proofs for Lambek-style logics, points in the canonical point set are taken to be formulas of the logic. This strategy won't work here, for the same reason given above in section 3, namely, that the polymorphism of lambda structures prevents using any single formula to represent an abstract. However, allowing points to be *sets* of formulas will suffice, following the general lines of the parametric proof of completeness for substructural logics given in Restall 2000 chapter 11.

Therefore let P, the set of points in the canonical frame, contain all non-empty sets of formulas that are closed under provability. That is, if we have some formula $A \in p$ for some point p, and $A \vdash B$, then $B \in p$. Then the subset relation $\subseteq$ will serve as the partial order $\leq$.

The accessibility relations for the canonical frame are defined as follows for all points a, b, and c, and for all formulas A and B:

$$R^\bullet abc \text{ iff } \forall AB.(A \in a \wedge B \in b) \rightarrow A \times B \in c$$
$$R^\circ abc \text{ iff } \forall AB.(A \in a \wedge B \in b) \rightarrow A \otimes B \in c$$

Clearly these accessibility relations are plump with respect to $\subseteq$.

Adding a canonical evaluation relation '$\Vdash$' defines the canonical model. For all points p and structures Δ,

$$p \Vdash \Delta \text{ iff } \{A \,|\, \Delta \vdash A \text{ is provable in NL}_\lambda\} \subseteq p$$

In particular, when Δ is a single formula A, given that points are closed under provability, this definition reduces to

$$p \Vdash A \text{ iff } A \in p$$

The accessibility relations and the evaluation relation work together to model provability in the logic.

Before proving completeness, we must establish that the canonical relation '$\Vdash$' respects the semantic constraints and the frame conditions.

Lemma 5 (respecting the semantics): The canonical model respects the semantic constraints.

Proof. We need to show that the canonical evaluation relation $\Vdash$ satisfies the general constraints on evaluation relations given above.

For atomic propositions A, $\Vdash$ is clearly upward monotonic with respect to $\leq$, since if $A \in p$ and $p \subseteq q$, then $A \in q$.

($\times$): The evaluation clause for '$\times$' is repeated here:

$$c \Vdash A \times B \text{ iff } \exists ab.R^\bullet abc \wedge a \Vdash A \wedge b \Vdash B$$

In the left to right direction, assume $c \Vdash A \times B$. Choose $a = \{A' \,|\, A \vdash A'\}$, and $b = \{B' \,|\, B \vdash B'\}$. Then $R^\bullet abc$ iff $\forall A' \in a, B' \in b. A' \times B' \in c$. We know that $A \times B \in c$ by assumption, and that $A \times B \vdash A' \times B'$ (use $\times L$ and $\times R$), and since c is closed under provability, $A' \times B' \in c$. The right to left direction follows immediately from the definition of $R^\bullet$.

($\otimes$): Similar to ($\times$).

($\backslash$): The evaluation clause for '$\backslash$' is repeated here:

$$b \Vdash A \backslash C \text{ iff } \forall ac.(R^\bullet abc \wedge a \Vdash A) \rightarrow c \Vdash C$$

In the left to right direction, assume $b \Vdash A \backslash C$. Choose arbitrary a and c, and assume $R^\bullet abc$ and $a \Vdash A$. Then $A \times (A \backslash C) \in c$ by the definition of $R^\bullet$, and since $A \times (A \backslash C) \vdash C$ and c is closed under provability, $c \Vdash C$. In the right to left direction, assume the right hand proposition holds. Choose $a = \{A' \,|\, A \vdash A'\}$, and let $c = \{C' \,|\, \exists A \in a, B \in b : A \times B \vdash C'\}$. Then $R^\bullet abc$ by the construction of c, so $C \in c$ by assumption. But also by the construction of c, there must exist some $A' \in a$ and $B \in b$ such that $A' \times B \vdash C$. By $\times L$ and cut, $A \bullet B \vdash C$, and so $B \vdash A \backslash C$. Since b is closed under provability, $b \Vdash A \backslash C$.

($/$), ($\backslash\!\!\backslash$), ($/\!\!/$): Similar to ($\backslash$).

We must also check that the canonical evaluation relation extends to structures in the appropriate way.

($\bullet$): Here is the evaluation condition for $\bullet$:

$$c \Vdash \Sigma \bullet \Delta \text{ iff } \exists ab.R^\bullet abc \wedge a \Vdash \Sigma \wedge b \Vdash \Delta$$

In the left to right direction, assume $c \Vdash \Sigma \bullet \Delta$. Choose $a = \{A \,|\, \Sigma \vdash A\}$ and $b = \{B \,|\, \Delta \vdash B\}$. Since $\Sigma \bullet \Delta \vdash A \times B$ for all $A \in a$ and for all $B \in b$ (use $\times R$), we have $R^\bullet abc$.

In the right to left direction, choose a and b as before, so $a = \{A \,|\, \Sigma \vdash A\}$ and $b = \{B \,|\, \Delta \vdash B\}$, and assume that $R^\bullet abc$. Choose an arbitrary C such that $\Sigma \bullet \Delta \vdash C$. By theorem 2, there exist σ and δ such that $\Sigma \vdash \sigma$ (which means $\sigma \in a$), $\Delta \vdash \delta$ (which means $\delta \in b$), and $\sigma \times \delta \vdash C$. Since $R^\bullet abc$, $\sigma \times \delta \in c$, so $C \in c$.

($\circ$): Similar to ($\bullet$).

(1): The evaluation clause for 1 says that for all i, $i \Vdash 1$ iff $i \Vdash t$. In the left to right direction, assume that $i \Vdash 1$. Since $1 \vdash t$, and i is closed under provability, $i \Vdash t$. In the right to left direction, assume $i \Vdash t$. Then for all A such that $1 \vdash A$, it follows that $t \vdash A$ by tL, and since points are closed under provability, $i \Vdash 1$.

(λ) The semantic clauses for lambda structures are repeated here:

$$c \Vdash \Sigma[\Delta] \text{ iff } \exists ab.R^\circ abc \wedge a \Vdash \Delta \wedge b \Vdash \lambda x \Sigma[x]$$
$$b \Vdash \lambda x \Sigma[x] \text{ only if } \forall ac\Delta.(R^\circ abc \wedge a \Vdash \Delta) \rightarrow c \Vdash \Sigma[\Delta]$$

For the first clause, for arbitrary $C \in c$, the bidirectionality of the abstraction postulate guarantees that $\Sigma[\Delta] \vdash C$ iff $\Delta \circ \lambda x \Sigma[x] \vdash C$. For the second clause, assume $b \Vdash \lambda x \Sigma[x]$. Choose arbitrary a, c, and Δ such that $a \Vdash \Delta$ and $R^\circ abc$. Then $c \Vdash \Delta \circ \lambda x \Sigma[x]$, so $c \Vdash \Sigma[\Delta]$. $\square$

Lemma 6 (respecting the frame conditions): The canonical model respects the frame conditions.

Proof. We need to show that the canonical model provides points that behave under the canonical evaluation relation as required by the frame conditions.

(left unit): Here is the frame constraint for the left unit:

$$\forall ab.a \leq b \leftrightarrow (\exists i.R^\bullet iab \wedge i \Vdash t)$$

Choose arbitrary a and b. In the left to right direction, assume $a \leq b$. Let $i = \{A \,|\, t \vdash A\}$, so we have $i \Vdash t$. Then $R^\bullet iab$ iff for all $I \in i$, $A \in a$, $I \times A \in b$. Since $a \leq b$, $A \in b$, and since b is closed under provability, $t \times A \in b$ and therefore $I \times A \in b$. In the right to left direction, assume that $R^\bullet iab$. Then $I \times A \in b$ for all $I \in i$ and $A \in a$. Since $I \times A \vdash A$, and b is closed under provability, $a \subseteq b$.

(right unit), (I): Similar to (left unit).

(B$^\bullet$): Here is frame condition (B$^\bullet$):

$$\forall abcd.(\exists e.R^\circ abe \wedge R^\bullet ecd) \leftrightarrow (\exists fgh.R^\circ cfg \wedge R^\circ bgh \wedge R^\circ ahd \wedge f \Vdash \lambda zyx((x \circ y) \bullet z))$$

Choose arbitrary a, b, c, and d.

In the left to right direction, assume there exists e such that $R^{\circ}abe$ and $R^{\bullet}ecd$. Choose arbitrary $A \in a$, $B \in b$, and $C \in c$. Then $(A \otimes B) \times C \in d$. Now let f, g, and h be the minimal sets that are closed under provability such that $f \Vdash \lambda zyx((x \circ y) \bullet z)$, $R^{\circ}cfg$, and $R^{\circ}bgh$. We need to show that $A \otimes (B \otimes (C \otimes F)) \in d$ for all $A, B, C, F \in a, b, c, f$. Since d is closed under provability, it will suffice to prove that $(A \otimes B) \times C \vdash A \otimes (B \otimes (C \otimes F))$. Starting with $A \circ (B \circ (C \circ F)) \vdash A \otimes (B \otimes (C \otimes F))$, since $\lambda zyx((x \circ y) \bullet z) \vdash F$ (by construction of f), a cut on F and three reductions completes the argument in the left to right direction.

In the right to left direction, assume there exists f, g, and h such that $f \vdash \lambda zyx((x \circ y) \bullet z)$, $R^{\circ}cfg$, $R^{\circ}bgh$, and $R^{\circ}ahd$. Choose e as the smallest set that is closed under provability and that contains $A \otimes B$ for all $A \in a$ and $B \in b$. Then we need to show that $(A \otimes B) \times C \in d$ for all A, B, C in a, b, c. Choose $F = C \backslash\!\backslash (B \backslash\!\backslash (A \backslash\!\backslash ((A \otimes B) \times C)))$. Clearly, $F \in f$, so $A \otimes (B \otimes (C \otimes F)) \in d$, and since d is closed under provability, $(A \otimes B) \times C \in d$.

(B$^{\circ}$), (C$^{\bullet}$), and (C$^{\circ}$): Similar to (B$^{\bullet}$). □

Theorem 4 (completeness): NL$_{\lambda}$ is complete with respect to the semantics and the frame conditions.

Proof. Lemmas 5 and 6 establish that the canonical model respects the semantics and the frame conditions. Assume $\Delta \vdash A$ is valid for some structure Δ and some formula A. Let d be a point in the canonical model such that $d = \{D \mid \Delta \vdash D\}$. Then $d \Vdash \Delta$. Since $\Delta \vdash A$ is valid, it follows that $d \Vdash A$. But $d \Vdash A$ iff $A \in d$. So by the construction of d, $\Delta \vdash A$ is provable in NL$_{\lambda}$. □

5 Relevant work

5.1 NL$_{\text{CL}}$

As mentioned in section 5, the soundness proof relies on ideas from the traditional embedding of lambda terms into combinatory logic. It should come as no surprise, then, that there is a combinatory version of the logic in which the abstraction postulate is replaced by the following five bidirectional structural postulates:

$$\frac{p}{\overline{p \circ \mathsf{I}}}\,\mathsf{I} \qquad \frac{(p \circ q) \bullet r}{\overline{p \circ (q \circ (r \circ \mathsf{B}^{\bullet}))}}\,\mathsf{B}^{\bullet} \qquad \frac{p \bullet (q \circ r)}{\overline{q \circ (r \circ (p \circ \mathsf{C}^{\bullet}))}}\,\mathsf{C}^{\bullet}$$

$$\frac{(p \circ q) \circ r}{\overline{p \circ (q \circ (r \circ \mathsf{B}^{\circ}))}}\,\mathsf{B}^{\circ} \qquad \frac{p \circ (q \circ r)}{\overline{q \circ (r \circ (p \circ \mathsf{C}^{\circ}))}}\,\mathsf{C}^{\circ}$$

Here, the structural symbols I, $\mathsf{B}^{\bullet}$, B°, $\mathsf{C}^{\bullet}$, and C° are zero-place structural connectives, that is, logical constants. See Barker 2007 and Barker and Shan 2014 for more details. This presentation is slightly different from the presentation in those works, which considered a restricted version of NL$_{\text{CL}}$. This version allows abstraction out of any context, just like the version of unrestricted NL$_{\lambda}$ studied in this paper.

The resemblance between these postulates and the frame conditions given above should be obvious. In some sense, these postulates fit the frame conditions more naturally. In fact, the strategy used in the works cited (and, incidentally, independently suggested for consideration by a referee) was to define NL_λ partly or entirely in terms of a mapping into $\mathrm{NL}_{\mathrm{CL}}$. As discussed in section 1, the goal of the present paper is to study NL_λ on its own terms, and to prove its metatheoretical properties directly. The advantages of NL_λ over $\mathrm{NL}_{\mathrm{CL}}$ include derivations that are much shorter and dramatically more understandable, as well as a much more natural correspondence with the traditional linguistic rule of quantifier raising.

5.2 The Displacement Calculus D

There have been a number of other substructural logics that have addressed scope-taking and movement, but undoubtedly the most closely relevant one is due to Morrill et al. 2011. They define a logic based on Lambek's associative logic *L* that they call the Displacement Calculus, D. There are both striking similarities and striking differences between their approach and the one discussed here.

As for the similarities, the correspondence is easiest to see at the level of formulas. Following Moortgat 1996, Morrill et al. define a type connective '$\uparrow$' 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 $A \backslash\!\!\backslash B$ here (note the reversal of the order of the subcategories). Likewise, they define a complementary connective '$\downarrow$' 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 $C /\!\!/ D$.

As for the differences, for one instance, D is not multimodal. As Morrill et al. put it (p. 13), "the displacement calculus has multimodal types but only unimodal sequents". Whether this seems like an advantage depends on theoretical taste, but it is equally legitimate to prefer the way that NL_λ cleanly separates local function/argument combination on the one hand and long-distance scope-taking and syntactic movement on the other into distinct modes of combination both syntactically and semantically.

In any case, models for *D* are built out of 'graded monoids', which are monoids with a homomorphism into the additive monoid of naturals. The sort or grade of an element in the model is the number of discontinuities it contains. Valentín 2012 proves that D is complete with respect to the class of general preordered displacement algebras. There is no need for grades in the semantics of NL_λ, of course; as we have seen, the semantics given here for NL_λ is a set of points with two (ungraded) monoid operations constrained by a set of frame conditions.

One intriguing property of the Displacement Calculus is that it contains no structural rules whatsoever. For instance, in addition to the treatment of scope-taking, Morrill et al.'s commitment to associativity, likewise their treatment of units, is "baked in" to the semantics and the logical rules.

But having fewer structural rules is not necessarily better. For instance, having a structural postulate for associativity (e.g., $p \bullet (q \bullet r) \equiv (p \bullet q) \bullet r$) lucidly and compactly captures the essence of associativity. For one advantage, the logic can easily be studied with and without associativity as desired, just by adding or removing the postulate (think of Lambek's NL and its associative version L).

In the same spirit, the abstraction postulate in NL_λ clearly and concisely expresses the essence of scope taking.

Surely having two such conceptually different but empirically convergent logical characterizations can only enrich our understanding of scope-taking and syntactic displacement.

6 Conclusions

NL$_\lambda$ is a Lambek-style substructural logic enriched by the addition of an unconventional structural inference ("abstraction"). Despite the unusual form of the abstraction inference, the logic is well-defined. Furthermore, it is well-behaved with respect to the usual properties expected of substructural logics, notably including cut elimination, interpolation, and decidability. In addition, NL$_\lambda$ is sound and complete with respect to the usual class of relational models for substructural logics.

Although the abstraction inference rule may be unconventional in comparison with standard structural rules, it closely resembles the traditional linguistic rule of Quantifier Raising, as well as the traditional notion of syntactic movement more generally. The many linguistic analyses of scope, binding, ellipsis, and movement that have been developed using NL$_\lambda$ suggest that it is indeed well-suited to linguistic theorizing. I conclude that NL$_\lambda$ is good candidate for the logic of scope taking and syntactic movement.

7 References

Barker, Chris. 2007. Parasitic scope. *Linguistics and Philosophy* **30.4**:407–444.

Barker, Chris. 2013. Scopability and sluicing. *Linguistics and Philosophy* **36.3**:187–223.

Barker, Chris. 2015a. Scope. In Shalom Lappin and Chris Fox (eds). *The Handbook of Contemporary Semantics*, 2d edition. Wiley-Blackwell. 47–87.

Barker, Chris. 2015b. Scope as Syntactic Abstraction. In Tsuyoshi Murata, Koji Mineshima, and Daisuke Bekki (eds). *New Frontiers in Artificial Intelligence*. 184–199.

Barker, Chris. In prep. The logic of movement. NYU manuscript.

Barker, Chris and Chung-chieh Shan. 2014, *Continuations and Natural Language*, Oxford University Press.

Buszkowski, Wojciech. 2005. Lambek Calculus with Nonlogical Axioms. In C. Casadio, P.J. Scott, and R.A.G. Seely (eds). *Language and Grammar. Studies in Mathematical Linguistics and Natural Language*. CSLI Lecture Notes volume 168. 77–93.

Buszkowski, Wojciech. 2010. Lambek Calculus and Substructural Logics. *Linguistic Analysis*, **36**:15–48.

Craig, William. 1957. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory *The Journal of Symbolic Logic* **22.3**: 269–285.

de Groote, Philippe. 2002. Towards abstract categorial grammars. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 148155. San Francisco, CA: Morgan Kaufmann.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell, Oxford.

Jäger, Gerhard. 2005. Residuation, structural rules, and context freeness. *Journal of Logic, Language, and Information* **13.1**: 49–57.

Kubota, Yusuke. 2015. Nonconstituent Coordination in Japanese as Constituent Coordination: An Analysis in Hybrid Type-Logical Categorial Grammar. *Linguistic Inquiry* **46.1**:1–42.

Lambek, Joachim. 1958. The mathematics of sentence structure. The *American Mathematical Monthly* **65**: 154–170.

Lambek, Joachim. 1961. On the calculus of syntactic types. In *Structure of Language and Its Mathematical Aspects*, edited by R. Jakobson, 166–178. Providence: American Mathematical Society.

Lambek, Joachim. 1988. Categorial and Categorical Grammars. In Richard T. Oehrle et al. (eds). *Categorial Grammars and Natural Language Structures*. 297–317. Reidel.

May, Robert. 1977. *The Grammar of Quantification*. Ph.D. thesis, MIT. Reprinted by New York: Garland, 1991.

May, Robert. 1985. *Logical Form: Its Structure and Derivation*. MIT Press, Cambridge.

Moortgat, Michael. 1997. Categorial type logics. In Johan F. A. K. van Benthem and Alice G. B. ter Meulen (eds). *Handbook of Logic and Language*. Elsevier. 93–178.

Morrill, Glyn, Oriol Valentín, and M. Fadda. 2011. The displacement calculus. *Journal of Logic, Language and Information* **20.1**: 1–48.

Muskens, Reinhard. 2001. λ-grammars and the syntax-semantics interface. In Robert van Rooy and Martin Stokhof (eds), *Proceedings of the 13th Amsterdam Colloquium*. Institute for Logic, Language and Computation, Universiteit van Amsterdam. 150–155.

Oehrle, Richard T. 1994. Term-labeled categorial type systems. *Linguistics and Philosophy* **17.6**:633–678.

Ono, Hiroakira. 2003. Substructural Logics and Residuated Lattices—an Introduction. In V.F. Hendricks and J. Malinowski (eds). *Trends in Logic: 50 Years of Studia Logica*. 193–228.

Pentus, Mati. 1993. Lambek grammars are context free. In *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science*: 429–433.

Restall, Greg. 2000. *An Introduction to Substructural Logics*. Routledge.

Roorda, Dirk. 1991. *Resource Logics: Proof-Theoretical Investigations*. PhD thesis, Amsterdam.

Vijay-Shankir, J., and David Wier. 1994. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory* **27.6**. 511–546.