

Strong and weak quantifiers in focused NL_{CL}

Wen Kokke

Institute for Logic, Language and Computation
University of Amsterdam, Amsterdam
wen.kokke@gmail.com

Abstract. We propose an improvement of Barker and Shan’s [4] NL_{CL} for which derivability is decidable, which has a normal-form for proof search, can analyse scope islands, and distinguish between strong and weak quantifiers.

Keywords: Categorical grammar · Focusing · Scope islands · Indefinite scope

1 Introduction

In 2014, Kiselyov and Shan [11] published a paper in which they presented an elegant approach to the analysis of various scope-related phenomena using, what they call, the continuation hierarchy. The phenomena they cover are scope ambiguity, scope islands and strong and weak quantifiers. They cover these phenomena using a mechanism which works on the sentence’s *semantics*, independent of whatever form of grammar is used.

At around the same time, Barker and Shan [4] published a book containing their findings on NL_λ and NL_{CL} , a pair of grammar logics, both with the ability to analyse scope ambiguity using a strictly *syntactic* mechanism. In addition, these logics can analyse “parasitic scope” [3, 4] and a quantifier which change the result type of the expressions they take scope over. However, neither of these logics is capable of analysing scope islands or strong and weak quantifiers.

In this paper, we rework NL_{CL} to a calculus which can analyse both scope islands and strong and weak quantifiers, without losing the ability to analyse parasitic scope or changing result types. For this, we base ourselves on work by Moortgat [13] and Moortgat and Moot [14]. This approach requires a strict focusing regime. Therefore, as an added bonus, adopting it results in the elimination of spurious ambiguity, and greatly enhances the efficiency of proof search when compared to Barker and Shan’s [4] NL_{CL} .

We will start our discussion by giving several examples of each of the aforementioned phenomena. The following sentences are examples of scope ambiguity, scope islands and weak quantifiers, respectively. They are given together with their expected semantics, and are based on examples by Szabolcsi [16, p. 608, 622].

- (1) “Someone read every book.”

- a. $\exists x.\mathbf{person}(x) \wedge \forall y.\mathbf{book}(y) \supset \mathbf{read}(x, y)$
- b. $\forall y.\mathbf{book}(y) \supset \exists x.\mathbf{person}(x) \wedge \mathbf{read}(x, y)$

(2) “Someone said Kurt wrote every book.”

- a. $\exists x.\mathbf{person}(x) \wedge \mathbf{say}(x, \forall y.\mathbf{book}(y) \supset \mathbf{wrote}(\mathbf{kurt}, y))$

(3) “Everyone said [Kurt dedicated a book to Mary].”

- a. $\forall x.\mathbf{person}(x) \supset \mathbf{say}(x, \exists y.\mathbf{book}(y) \wedge \mathbf{dedicate}(\mathbf{kurt}, \mathbf{mary}, y))$
- b. $\forall x.\mathbf{person}(x) \supset \exists y.\mathbf{book}(y) \wedge \mathbf{say}(x, \mathbf{dedicate}(\mathbf{kurt}, \mathbf{mary}, y))$
- c. $\exists y.\mathbf{book}(y) \wedge \forall x.\mathbf{person}(x) \supset \mathbf{say}(x, \mathbf{dedicate}(\mathbf{kurt}, \mathbf{mary}, y))$

The first of these examples is a canonical example of scope ambiguity. Example (2) demonstrates a scope island: there is no reading in which “every book” scopes out of the embedded clause, as this reading would imply that there was potentially a different speaker for each book—“Alex said Kurt wrote Slaughterhouse-Five”, “Jules said Kurt wrote Cat’s Cradle”, “Sam said Kurt wrote. . .” Example (3) shows that indefinites *can* scope out of scope islands.

We add two more sentences, which are examples of a quantifier which changes the result type, and of parasitic scope, respectively. These examples based on those given by Barker and Shan’s [4, p. 208] and Kiselyov [10].

(4) “John read a book [the author of which] feared the ocean.”

- a. $\exists x.\mathbf{book}(x) \wedge \mathbf{fear}(\iota(\lambda y.\mathbf{of}(y, \mathbf{author}, x)), \iota(\mathbf{ocean})) \wedge \mathbf{read}(\mathbf{john}, x)$

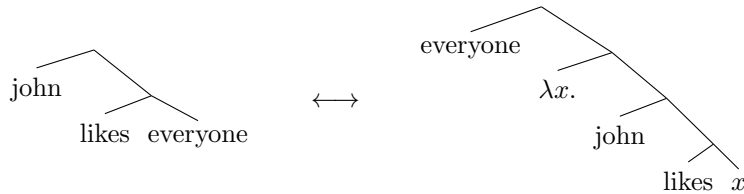
(5) “Everyone feared the same ocean.”

- a. $\exists z.\forall y.\mathbf{fear}(y, \iota(\lambda x.\mathbf{ocean}(x) \wedge x = z))$

These last two examples will play a less important role, as NL_{CL} is already capable of analysing both. However, in order to demonstrate that we have not lost that capability, we will provide analyses of both near the end of this paper.

2 Background

In this section, we will briefly discuss NL_{CL} and its sibling, NL_{λ} . NL_{CL} is an extension to the non-associative Lambek calculus [NL; 12]. The history behind NL_{CL} is somewhat intricate, but helpful to understanding, so we will briefly go over it. The initial idea comes from the practice of encoding quantifier movement as a tree transformation which introduces a binder [9]:



To implement this idea in type-logical grammar, Barker and Shan add a structural λ -construct to NL, and added the following structural postulate:¹

$$\Sigma[\Gamma] \longleftrightarrow \Gamma \circ \lambda x. \Sigma[x] \quad (\lambda)$$

As can be seen, the (λ) postulate uses a new connective: the \circ (hollow product). This connective is part of a new residuated family $\{\backslash, \circ, /\}$, which starts out as a copy of $\{\backslash, \bullet, /\}$. However, the addition of the (λ) postulate allows you to raise any constituent to the top-left² position in the structure, where—if it has the right type—it can be “resolved” against the top-level type as follows:

$$\frac{\frac{\frac{\Sigma[A] \vdash B}{A \circ \lambda x. \Sigma[x] \vdash B} (\lambda)}{\lambda x. \Sigma[x] \vdash A \backslash B} R\backslash}{\frac{C \backslash (A \backslash B) \circ \lambda x. \Sigma[x] \vdash D}{\Sigma[C \backslash (A \backslash B)] \vdash D} (\lambda)} C \vdash D \quad L/\quad L/\quad L/$$

Barker and Shan call resulting system NL_λ . While NL_λ fulfils the promise of allowing a syntactic analysis of quantifier raising, scope ambiguity and parasitic scope, it has some problems. Most notably, the system is hard to formalise and to reason about, largely due to the presence of a binding construct in the syntax of structures. While it is not impossible to formalise, the (λ) postulate greatly complicates meta-logical proofs.

To address this issue, and to ease their own investigation of the formal properties of NL_λ , Barker and Shan [4, ch. 17] introduce NL_{CL} . This system uses the fact that λ -terms can be represented as combinators in combinatory logic, which removes the need for a binding construct. Barker and Shan use a variant of Schönfinkel’s mapping to encode the *linear* λ -construct as applications of the combinators **I**, **B** and **C**:³⁴

$$\mathbf{I}x = x, \quad \mathbf{B}xyz = x(yz), \quad \mathbf{C}xyz = xzy$$

The resulting system is presented in figure 1.⁵

Using the system in figure 1, we can do quantifier raising in much the same way as we did with the (λ) postulate—although, as we now have to raise the quantifier one step at a time, the proofs are much longer:

¹ It is important to note that this construct is *purely structural*, and that it is not accompanied by some implicit form of computation (e.g. β, η -conversions).

² It should be noted that the decision to raise quantifiers to the top-*left* position, as opposed to the top-*right*, is a stylistic choice made by Barker and Shan [4]. It is entirely possible to use the mirrored versions of the **IBC**-rules together with the $(B \backslash A) \backslash C$ type for quantifiers.

³ One can easily verify that the λ -construct introduced by (λ) is linear.

⁴ When comparing these equations to the **IBC**-rules in figure 1, note that \bullet encodes function application, but \circ encodes *flipped* function application.

⁵ In figure 1, and for the remainder of this paper, the letters Γ and Δ are reserved for structures, whereas the Σ is used for contexts.

⁶ When reading this figure, be wary of the difference between the combinators **B**, **C** and the formulas B, C .

$$\begin{array}{c}
\overline{A \vdash A} \text{ Ax} \\
\\
\frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[\Gamma \bullet A \setminus B] \vdash C} \setminus L \quad \frac{A \bullet \Gamma \vdash B}{\Gamma \vdash A \setminus B} \setminus R \quad \frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[B / A \bullet \Gamma] \vdash C} /L \quad \frac{\Gamma \bullet A \vdash B}{\Gamma \vdash B / A} /R \\
\\
\frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[\Gamma \circ A \setminus B] \vdash C} \setminus L \quad \frac{A \circ \Gamma \vdash B}{\Gamma \vdash A \setminus B} \setminus R \quad \frac{\Gamma \vdash A \quad \Sigma[B] \vdash C}{\Sigma[B \not\! / A \circ \Gamma] \vdash C} \not\! /L \quad \frac{\Gamma \circ A \vdash B}{\Gamma \vdash B \not\! / A} \not\! /R \\
\\
\frac{\Sigma[A] \vdash B}{\Sigma[A \circ \mathbf{I}] \vdash B} \mathbf{I} \quad \frac{\Sigma[A \bullet (B \circ C)] \vdash D}{\Sigma[B \circ ((\mathbf{B} \bullet A) \bullet C)] \vdash D} \mathbf{B} \quad \frac{\Sigma[(A \circ B) \bullet C] \vdash D}{\Sigma[A \circ ((\mathbf{C} \bullet B) \bullet C)] \vdash D} \mathbf{C}
\end{array}$$

Fig. 1. NL_{CL} as presented by Barker and Shan [4].⁶

$$\begin{array}{c}
\vdots \\
\frac{\text{JOHN} \bullet \text{LIKES} \bullet np \vdash s}{\text{JOHN} \bullet \text{LIKES} \bullet np \circ \mathbf{I} \vdash s} \mathbf{I} \\
\frac{\text{JOHN} \bullet np \circ (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s}{np \circ (\mathbf{B} \bullet \text{JOHN}) \bullet (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s} \mathbf{B} \\
\frac{np \circ (\mathbf{B} \bullet \text{JOHN}) \bullet (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s}{(\mathbf{B} \bullet \text{JOHN}) \bullet (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash np \not\! / s} \not\! /R \quad \frac{s \vdash s}{\text{EVERYONE} \circ (\mathbf{B} \bullet \text{JOHN}) \bullet (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s} \not\! /L \\
\frac{\text{EVERYONE} \circ (\mathbf{B} \bullet \text{JOHN}) \bullet (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s}{\text{JOHN} \bullet \text{EVERYONE} \circ (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s} \mathbf{B} \\
\frac{\text{JOHN} \bullet \text{EVERYONE} \circ (\mathbf{B} \bullet \text{LIKES}) \bullet \mathbf{I} \vdash s}{\text{JOHN} \bullet \text{LIKES} \bullet \text{EVERYONE} \circ \mathbf{I} \vdash s} \mathbf{B} \\
\frac{\text{JOHN} \bullet \text{LIKES} \bullet \text{EVERYONE} \circ \mathbf{I} \vdash s}{\text{JOHN} \bullet \text{LIKES} \bullet \text{EVERYONE} \vdash s} \mathbf{I}
\end{array}$$

The labels JOHN, LIKES and EVERYONE abbreviate the types np , $(np \setminus s) / np$ and $s \not\! / (np \not\! / s)$, respectively. For a more detailed account of the relation between NL_λ and NL_{CL} , see Barker and Shan [4]. For a more detailed account of various encodings of combinatorial logic in structural rules, amongst which the encoding of the linear lambda construct used by Barker and Shan, see Finger [8].

3 Scope Islands for NL_{CL}

Our aim for this section is to present an extension to NL_{CL} which will allow us to analyse scope islands, and therefore example (2).

To analyse scope islands, we need some way to block quantifier movement. If you look at the **IBC**-rules in figure 1, you will notice that they allow constituents attached to (the left of) a hollow product to move past solid products. This leads us to suggest a fairly simple solution: insert *anything* that is not solid product. For this, we use a residuated pair of unary connectives, \diamond and \square [15, 13]. The relevant rules are presented in figure 2.

$$\frac{\Sigma[A] \vdash B}{\Sigma[(\square A)] \vdash B} \square L \quad \frac{\langle \Gamma \rangle \vdash B}{\Gamma \vdash \square B} \square R \quad \frac{\Sigma[(\langle A \rangle)] \vdash B}{\Sigma[\langle \diamond A \rangle] \vdash B} \diamond L \quad \frac{\Gamma \vdash B}{\langle \Gamma \rangle \vdash \diamond B} \diamond R$$

Fig. 2. Scope islands for NL_{CL} .

Using these connectives, we can assign ‘said’ the type $(np \setminus s) / \diamond s$. Instead of taking a sentence-argument from the right, ‘said’ now takes a *closed-off* sentence—a scope island. Have a look at the derivation for example (2) given below:

$$\frac{\frac{\frac{\vdots}{\text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \vdash s}{} \diamond R}{(\text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK}) \vdash \diamond s} \quad \frac{\frac{\vdots}{\text{SOMEONE} \bullet (np \setminus s) \vdash s}{} /L}{\text{SOMEONE} \bullet \text{SAID} \bullet (\text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK}) \vdash s}}{} /L$$

As long as the scope island (written $\langle \cdot \rangle$) is in place, ‘EVERY • BOOK’ cannot be raised past it, for there is no rule which allows anything to move past a diamond. But in order to remove the scope island, it has to be eliminated against the $\diamond s$ argument of ‘said’, and doing so isolates the embedded clause in its own branch of the proof.⁷

4 Strong and Weak Quantifiers

In the previous section, we presented an extension to NL_{CL} which enabled us to analyse scope islands. This extension blocks all quantifier movement out of scope islands. Example (3) demonstrates that this is too coarse an approach. Specifically, we would like to allow weak quantifiers, such as indefinites, to scope out of scope islands.

We could approach this issue as a syntactic problem, and encode it using structural rules, as we did with quantifier movement and scope islands.⁸ However, Szabolcsi [16] writes that “indefinites acquire their existential scope in a manner that does not involve movement and is essentially syntactically unconstrained.” Therefore, we feel that a syntactic approach would be out of place.

How do we approach the problem of weak quantifiers as a semantic problem? The solution is to use continuation-passing style (CPS). But how? Early attempts, such as the work by [2], often works by applying a CPS translation directly to the semantic terms. Such approaches, however, face a fundamental dilemma. Because the CPS translation is applied to a solitary semantic term, a deterministic translation cannot introduce scope ambiguity—or any ambiguity, for that matter. However, making the CPS translation sufficiently nondeterministic without causing spurious ambiguity is an arduous task. When Barker makes the translation ambiguous, in order to capture scope ambiguity, this

⁷ The presence of the structural diamond in the endsequent may seem problematic, but recall that from the perspective of backward-chaining search we assign semantics to a *known* sentence structure. If we switch to forward-chaining search, i.e. to parsing, the need for a scope island will be inferred from the type of ‘said’.

⁸ For instance, we could split the family $\setminus, \circ, //$ into two separate families, $\setminus^w, \circ^w, //^w$ and $\setminus^s, \circ^s, //^s$, each with their own copies of the **IBC**-rules, and add a structural rule which selectively allows weak quantifiers to move past scope islands.

leads to the number of introduced ambiguous interpretations growing exponentially with the sentence length. More recent approaches, such as the work by Kiselyov and Shan [11], are much more sophisticated. Their approach allows for the creation of quantifiers of different strengths (e.g. everyone_1 , everyone_2 , ...) essentially reducing scope ambiguity to lexical ambiguity. As a linguistic standpoint, this feels wrong. Furthermore, their framework was engineered to be able to analyse phenomena such as scope islands and weak quantifiers. This makes it too expressive (and intricate) for the task at hand.

Instead, we base our CPS semantics on the approach of Moortgat and Moot [14] and Bastenhof [6], who manage to elegantly integrate CPS semantics into their grammar logic. Moortgat and Moot set up a calculus which enforces one crucial property: every proof in the grammar logic is associated with *unique, normal-form semantics*. In the context of scope ambiguity, this means that each way to interpret a sentence with ambiguous scope corresponds to exactly one proof in the grammar logic.

Focused NL_{CL} Moortgat and Moot [14, sec. 3.1] define a normal-form calculus for the Lambek-Grishin calculus (LG). They refer to this calculus as fLG—for *focused* LG, after the technique, pioneered by Andreoli [1], which they use in their calculus. Their version of focusing, however, is more general than that of Andreoli, as they allow for the arbitrary assignment of polarities to atoms. Andreoli’s [1] schema can be recovered by assigning all atomic formulas *negative* polarity.

As NL is a fragment of LG, we can trivially extract a normal-form calculus for NL from their work. We will, in their style, refer to this calculus as fNL.

It is important to note that they develop their calculus within the framework of display calculus [7]. One advantage of this framework is that we can freely add structural rules, without fear that we will lose the cut-elimination property. Barker and Shan’s [4] extension of NL, NL_{CL} , consists solely of a copy of an existing modality ($\backslash, \circ, /$) and a number of structural rules. Therefore, by applying these same changes, we can extend fNL to focused NL_{CL} —or f NL_{CL} . The result is presented in figure 3, together with the focused version of the extension for scope islands from section 3.

Equivalence between f NL_{CL} and NL_{CL} can likely be proven using an intermediate system: display NL_{CL} . One can trivially obtain this system from the focused system in figure 3 by dropping the focus marker “ \square ” and the focusing and unfocusing rules. Equivalence between the display and focused variants of a system was proven for classical NL by Bastenhof [5]. This proof can likely be adapted for NL_{CL} .

However, it is important to realise that, even in the absence of a formal proof of equivalence between NL_{CL} and f NL_{CL} , the second remains a logical

system which can analyse all phenomena which Barker and Shan [4] show NL_{CL} can analyse.⁹

Decidable Proof Search At this point, fNL_{CL} still has a problem, which it shares with NL_{CL} : we do not have a decidable procedure for proof search. Since it is a grammar logic, this means that we do not have a procedure for *parsing*. An easy way to obtain such a procedure is to change the system in such a way that backward-chaining search becomes decidable. The reason this is not decidable in NL_{CL} is because of the **I**-rule, which does not obey the substructure property.¹⁰

Admittedly, there are other rules which do not obey the substructure property: the residuation rules and the **B** and **C** rules do not enjoy it. However, the residuation rules still enjoy a weak form of this property: they *do not increase* the size of the structure. This means that we can use loop checking to filter out problematic branches of the search. More interestingly, the **B** and **C** rules have the property that “whatever goes up, must come down.” At some point, the quantifier will reach the top of the expression, and at that point, there are only two things to do: (1) resolve the quantifier against the top-level type, thereby *eliminating* a connective and breaking out of any loop; or (2) go back down along the same path. Yet when searching for a proof with the **I**-rule, we can always introduce another **I**.

We will address this issue by restricting access to the **I**-rule using a license. This license will be a new unary connective, written **QA**. Semantically, this logical connective corresponds to a hollow product with a right-hand **I** (i.e. $A \circ \mathbf{I}$). However, as we want neither hollow products, nor the unit **I**, on the logical level, we capture these in a single connective. We remove the **I**-rule, and add the following three rules to the calculus in figure 3:

$$\frac{\cdot A \cdot \circ \mathbf{I} \vdash \Delta}{\cdot \mathbf{QA} \cdot \vdash \Delta} \mathbf{IL} \quad \frac{\Gamma \vdash \boxed{B}}{\Gamma \circ \mathbf{I} \vdash \boxed{\mathbf{QB}}} \mathbf{IR} \quad \frac{\Gamma \vdash \Delta}{\Gamma \circ \mathbf{I} \vdash \Delta} \mathbf{I}^-$$

The first two of these rules are the display calculus rules for right-hand products. The third is the remaining direction of the original **I**-rule. With this change, quantifier raising is restricted to expressions of the form $\mathbf{Q}(C \int (A \int B))$, and proof search becomes decidable.

One problem which remains is that the **B** and **C** rules cause a huge amount of spurious ambiguity. To see why, note that when raising multiple quantifiers, it is possible to intersperse the various applications of the **B** and **C** rules in many different ways. To solve this, we will take some inspiration from Barker and Shan [4, ch. 17.6], who solve this issue, albeit in a convoluted way. They show that NL_λ can be embedded in NL_{CL} , using a variant of Schönfinkel’s

⁹ Throughout the remainder of the paper, whenever we discuss one of the phenomena discussed in section 1, we will give an example proof in fNL_{CL} . In this way, by the end of this paper, this claim will be backed up by evidence.

¹⁰ In the case of NL_{CL} , this is the subformula property.

mapping from λ -terms to combinatory logic. Later, they show that a pair of derived rules, $\llcorner L_\lambda$ and $\lrcorner R_\lambda$, can serve as a normal-form for the structural rules of NL_{CL} . However, these derived rules employ the structural λ which, in the context of NL_{CL} , is presumably immediately translated using Schönfinkel's mapping. Instead of employing this two-step process, we exploit the similarities between single-hole contexts and linear λ -terms to derive a variant of the λ -rule which directly uses Schönfinkel's mapping (written \cdot) [cf. 4, ch. 17.5]:

$$\begin{aligned} \bar{\square} &= \mathbf{I} \\ \frac{\cdot A \cdot \circ \bar{\Sigma} \vdash \Delta}{\Sigma \bullet \Gamma} &= ((\mathbf{C} \bullet \bar{\Sigma}) \bullet \Gamma) & \frac{\cdot A \cdot \circ \bar{\Sigma} \vdash \Delta}{\Sigma[\mathbf{QA}] \vdash \Delta} \updownarrow \\ \frac{\cdot A \cdot \circ \bar{\Sigma} \vdash \Delta}{\Gamma \bullet \Sigma} &= ((\mathbf{B} \bullet \Gamma) \bullet \bar{\Sigma}) \end{aligned}$$

We can use this mapping in the definition of a derived rule: the \updownarrow -rule, written as \uparrow or \downarrow , depending on the direction in which it is applied.¹¹ We can derive the this rule using three lemmas:

$$\mathbf{Q/I}: \frac{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta}{\Sigma[\mathbf{QA}] \vdash \Delta} \mathbf{Q/I} \quad \mathbf{I}^{-'}: \frac{\Sigma[\Gamma] \vdash \Delta}{\Sigma[\Gamma \circ \mathbf{I}] \vdash \Delta} \mathbf{I}^{-'} \quad \updownarrow': \frac{\Gamma \circ \Sigma[\Gamma'] \vdash \Delta}{\Sigma[\Gamma \circ \Gamma'] \vdash \Delta} \updownarrow'$$

Using these lemmas, we can derive the two directions of \updownarrow as follows:

$$\uparrow: \frac{\cdot A \cdot \circ \bar{\Sigma} \vdash \Delta}{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta} \updownarrow' \quad \downarrow: \frac{\Sigma[\cdot A \cdot] \vdash \Delta}{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta} \mathbf{I}^{-'} \quad \updownarrow': \frac{\Sigma[\cdot A \cdot] \vdash \Delta}{\cdot A \cdot \circ \bar{\Sigma} \vdash \Delta} \updownarrow'$$

The lemmas themselves can be derived by induction on the structure of the context Σ . The derivation of $\mathbf{Q/I}$ and $\mathbf{I}^{-'}$ is done as follows:

$$\begin{aligned} \square: \frac{\cdot A \cdot \circ \mathbf{I} \vdash \Delta}{\cdot \mathbf{QA} \cdot \vdash \Delta} \mathbf{IL} \quad \Sigma \bullet \Gamma: \frac{\frac{\Sigma[\cdot A \cdot \circ \mathbf{I}] \bullet \Gamma \vdash \Delta}{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta / \Gamma} \text{Res}/\bullet}{\frac{\Sigma[\mathbf{QA}] \vdash \Delta / \Gamma}{\Sigma[\mathbf{QA}] \bullet \Gamma \vdash \Delta} \text{Res}/\bullet} \mathbf{Q/I} \quad \Gamma \bullet \Sigma: \frac{\frac{\Gamma \bullet \Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta}{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Gamma \setminus \Delta} \text{Res}\backslash\bullet}{\frac{\Sigma[\mathbf{QA}] \vdash \Gamma \setminus \Delta}{\Gamma \bullet \Sigma[\mathbf{QA}] \vdash \Delta} \text{Res}\backslash\bullet} \mathbf{Q/I} \\ \square: \frac{\cdot A \cdot \vdash \Delta}{\cdot A \cdot \circ \mathbf{I} \vdash \Delta} \mathbf{I}^{-'} \quad \Sigma \bullet \Gamma: \frac{\frac{\Sigma[\cdot A \cdot] \bullet \Gamma \vdash \Delta}{\Sigma[\cdot A \cdot] \vdash \Delta / \Gamma} \text{Res}/\bullet}{\frac{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta / \Gamma}{\Sigma[\cdot A \cdot \circ \mathbf{I}] \bullet \Gamma \vdash \Delta} \text{Res}/\bullet} \mathbf{I}^{-'} \quad \Gamma \bullet \Sigma: \frac{\frac{\Gamma \bullet \Sigma[\cdot A \cdot] \vdash \Delta}{\Sigma[\cdot A \cdot] \vdash \Gamma \setminus \Delta} \text{Res}\backslash\bullet}{\frac{\Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Gamma \setminus \Delta}{\Gamma \bullet \Sigma[\cdot A \cdot \circ \mathbf{I}] \vdash \Delta} \text{Res}\backslash\bullet} \mathbf{I}^{-'} \end{aligned}$$

These rules simply introduce or eliminate the unit \mathbf{I} under some context Σ . The actual movement takes place in the definition of \updownarrow' . In this proof, the base case is simply the identity, as no movement is required to move out of the empty context:

$$\Sigma \bullet \Gamma: \frac{\frac{\frac{\Gamma \circ ((\mathbf{C} \bullet \Sigma[\Gamma']) \bullet \Gamma'') \vdash \Delta}{(\Gamma \circ \Sigma[\Gamma']) \bullet \Gamma'' \vdash \Delta} \mathbf{C}}{\frac{\Gamma \circ \Sigma[\Gamma'] \vdash \Delta / \Gamma''}{\Sigma[\Gamma \circ \Gamma'] \vdash \Delta / \Gamma''} \text{Res}/\bullet} \updownarrow'}{\frac{\Sigma[\Gamma \circ \Gamma'] \vdash \Delta / \Gamma''}{\Sigma[\Gamma \circ \Gamma'] \bullet \Gamma'' \vdash \Delta} \text{Res}/\bullet} \updownarrow' \quad \Gamma \bullet \Sigma: \frac{\frac{\frac{\Gamma \circ ((\mathbf{B} \bullet \Gamma'') \bullet \Sigma[\Gamma']) \vdash \Delta}{\Gamma'' \bullet (\Gamma \circ \Sigma[\Gamma']) \vdash \Delta} \mathbf{B}}{\frac{\Gamma \circ \bar{\Sigma} \vdash \Gamma'' \setminus \Delta}{\Sigma[\Gamma \circ \Gamma'] \vdash \Gamma'' \setminus \Delta} \text{Res}\backslash\bullet} \updownarrow'}{\frac{\Gamma'' \bullet \Sigma[\Gamma \circ \Gamma'] \vdash \Delta}{\Gamma'' \bullet \Sigma[\Gamma \circ \Gamma'] \vdash \Delta} \text{Res}\backslash\bullet} \updownarrow'$$

¹¹ These correspond to Barker and Shan's [4, p. 201] EXPANSION and REDUCTION rules, respectively.

Note that the \uparrow -rule eliminates a logical connective—the **Q**—and therefore has the subformula property. In addition, the \downarrow -rule, on the other hand, eliminates the trail of **B**s and **C**s, and thus has the substructure property. Because of this, proof search with these rules is decidable.

Furthermore, proof search with the $\uparrow\downarrow$ -rule is complete. Briefly, this is true because the **IBC**-rules can do nothing but move a constituent up, or down along an existing path—the $\uparrow\downarrow$ -rule mere captures this more succinctly. A formal proof of this can be given by implementing a normalisation function using the commutative conversions for the **B** and **C** rules: one can move the applications of the **B** and **C** rules around until they form a continuous sequence (interspersed with residuation rules) starting (or ending) with an application of the **I**-rule. This sequence of applications can then be replaced by a single application of the $\uparrow\downarrow$ -rule. Therefore, proof search using the $\uparrow\downarrow$ -rule is complete with respect to the the **IBC**-rules.

We follow Barker and Shan [4], and derive rules corresponding to the $\uparrow\downarrow L_\lambda$ - and $\uparrow\downarrow R_\lambda$ -rules. These rules combine an application of $\uparrow\downarrow$ -rule with an application of $\uparrow\downarrow L$ or $\uparrow\downarrow R$. We name them qL and qR , to signify that they no longer employ a structural λ , and because they can be composed to implement Moortgat’s [13] q -connective:

$$\begin{array}{c}
 \frac{\overline{\Sigma} \vdash \boxed{A \backslash B} \quad \boxed{C} \vdash \Delta}{\boxed{C \uparrow (A \backslash B)} \vdash \Delta \uparrow \overline{\Sigma}} \uparrow L \\
 \frac{\overline{\Sigma} \vdash \boxed{A \backslash B} \quad \boxed{C} \vdash \Delta}{\boxed{C \uparrow (A \backslash B)} \vdash \Delta \uparrow \overline{\Sigma}} \uparrow L \\
 qL: \frac{\frac{\frac{\overline{\Sigma} \vdash \boxed{A \backslash B} \quad \boxed{C} \vdash \Delta}{\boxed{C \uparrow (A \backslash B)} \vdash \Delta \uparrow \overline{\Sigma}} \uparrow L \quad \text{Foc}^L}{\cdot C \uparrow (A \backslash B) \cdot \vdash \Delta \uparrow \overline{\Sigma}} \text{Foc}^L}{\cdot C \uparrow (A \backslash B) \cdot \circ \overline{\Sigma} \vdash \Delta} \text{Res} \uparrow \circ}{\overline{\Sigma} \vdash \mathbf{Q}(C \uparrow (A \backslash B)) \cdot \vdash \Delta} \uparrow
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\Sigma[A] \vdash \cdot B \cdot}{\cdot A \cdot \circ \overline{\Sigma} \vdash \cdot B \cdot} \downarrow \\
 qR: \frac{\cdot A \cdot \circ \overline{\Sigma} \vdash \cdot B \cdot}{\overline{\Sigma} \vdash \cdot A \cdot \backslash \cdot B \cdot} \text{Res} \circ \downarrow}{\overline{\Sigma} \vdash \cdot A \cdot \backslash B \cdot} \backslash R
 \end{array}$$

As these rules eliminate at least one logical connective each, they still enjoy the subformula property, so proof search with these rules is decidable. In fact, it is *slightly* more efficient than with the $\uparrow\downarrow$ -rule. The reason for this is that after raising a quantifier, the only course of action is applying the $\uparrow\downarrow L$ -rule anyway—and likewise for qR .¹²

Henceforth, if we refer to proof search for fNL_{CL} , we are referring to search using the logical and residuation rules for $\backslash, \bullet, /, \backslash, \circ, \uparrow$ and \diamond, \square , and the qL and qR rules.¹³

Continuation Semantics for NL_{CL} A normal-form calculus for proof search is a great improvement, but we were really after Moortgat and Moot’s [14, sec. 3.1] CPS semantics. As with the calculus itself, we can trivially restrict their translation function to fNL , and then extend it to cover fNL_{CL} . In figure 4, we present the translation on types and sequents.

¹² This can be proven using a variant of Barker and Shan’s [4, ch. 17.6 and 17.7] proof of equivalence between NL_λ and NL_{CL} .

¹³ In fact, no proof will ever explicitly use the logical or residuation rules for $\backslash, \circ, \uparrow$, leading to a question of whether it is really necessary for \uparrow and \backslash to be fully residuated logical implications. But this is a matter for another paper.

We extend the translation on types to a translation on structures as follows: we translate all structural constants (**I**, **B**, **C**) as units, forget all unary structural connectives (\diamond , \square), and translate all binary structural connectives as products. Atomic structures $\cdot A$ are translated as $\llbracket A \rrbracket^-$ or $\llbracket A \rrbracket^+$, depending on the polarity of the structure $\cdot A$.

In this particular CPS translation, all function applications and abstractions are contained within the focusing and unfocusing rules, which are translated as follows:

$$\frac{x : \llbracket \Delta \rrbracket \vdash M : \llbracket A \rrbracket^-}{k : \llbracket A \rrbracket^+ \vdash (\lambda x.k M) : \llbracket \Delta \rrbracket^R} \text{Foc}^L \quad \frac{x : \llbracket A \rrbracket^+ \vdash M : \llbracket \Delta \rrbracket^R}{y : \llbracket \Delta \rrbracket \vdash (\lambda x.M y) : \llbracket A \rrbracket^-} \text{Unf}^L$$

$$\frac{x : \llbracket \Gamma \rrbracket \vdash M : \llbracket A \rrbracket^+}{x : \llbracket \Gamma \rrbracket \vdash (\lambda k.k M) : \llbracket A \rrbracket^{-R}} \text{Foc}^R \quad \frac{x : \llbracket \Gamma \rrbracket \vdash M : \llbracket A \rrbracket^{-R}}{x : \llbracket \Gamma \rrbracket \vdash (\lambda y.M y) : \llbracket A \rrbracket^+} \text{Unf}^R$$

The other rules are translated either as axioms (Ax^L , Ax^R), identities ($\backslash R$, $/R$, $\backslash\backslash R$, $//R$ and all rules for \diamond , \square) or permutations (the rest). For instance,

$$\frac{x : \llbracket \Gamma \rrbracket \vdash M : \llbracket A \rrbracket^+ \quad y : \llbracket \Delta \rrbracket \vdash N : \llbracket B \rrbracket^-}{z : \llbracket \Gamma \rrbracket \times \llbracket \Delta \rrbracket \vdash (M[\pi_1 z/x], N[\pi_2 z/y]) : \llbracket A \rrbracket^+ \times \llbracket B \rrbracket^-} \backslash L$$

An exception to this are the **I**-rules. Because we would like to be able to simply forget the **Q**-connective upon translation, so that we do not have to store unnecessary units in our lexicon, we have to insert or remove the units upon using these rules.

Using these semantics, we can assign the indefinite article the type np/n .¹⁴ This will result in *two* interpretations for (2), and *three* interpretations for (3), as required. Let us consider the important steps in the derivation of (3):

1. the quantifier movement and scope taking of “everyone”;
2. the collapsing of the scope island, isolating the clause “[_S Kurt .. Mary]” in its own branch of the derivation;
3. the collapsing of “a book”, with the indefinite taking scope at the top-level.

If these steps are taken [1,2,3], we obtain interpretation (3a); if they are taken [1,3,2], we obtain (3b); and if they are taken [3,1,2], we obtain (3c).¹⁵

5 Examples

In this section, we will present a number of analyses of the examples presented in section 1. In the interest of brevity, we will summarise numerous applications of the residuation rules, beginning or ending with focusing or unfocusing rules with ‘dp’, for *display postulate*. In addition, we will leave out uninteresting subproofs.

¹⁴ Quantifiers such as “someone” should be assigned the type $np/n \otimes n$, which means we must also extend NL_{CL} with logical products.

¹⁵ The normal-form requires that 1 occurs before 2, so this list is exhaustive.

First off, we present an analysis of (1), resulting in interpretation (1b). The quantifier EVERY is assigned the type $\mathbf{Q}(s \parallel (np \setminus s)) / n$, and SOMEONE is assigned the type of a “strong” quantifier—that is to say, $\mathbf{Q}(s \parallel (np \setminus s))$.

$$\begin{array}{c}
\vdots \\
\frac{\cdot np \cdot \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot s}{\square \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot np \setminus s} qR \\
\frac{\square \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot np \setminus s}{\square \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot \boxed{np \setminus s}} UnfR \\
\frac{\square \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot \boxed{np \setminus s}}{\boxed{s} \vdash \cdot s} AxL \\
\frac{\boxed{s} \vdash \cdot s}{\square \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot \boxed{np \setminus s}} qL \\
\frac{\text{SOMEONE} \bullet \text{READ} \bullet \cdot np \cdot \vdash \cdot s}{\text{SOMEONE} \bullet \text{READ} \bullet \square \vdash \cdot np \setminus s} qR \\
\frac{\text{SOMEONE} \bullet \text{READ} \bullet \square \vdash \cdot np \setminus s}{\text{SOMEONE} \bullet \text{READ} \bullet \square \vdash \cdot \boxed{np \setminus s}} UnfR \\
\frac{\text{SOMEONE} \bullet \text{READ} \bullet \square \vdash \cdot \boxed{np \setminus s}}{\boxed{s} \vdash \cdot s} AxL \\
\frac{\text{SOMEONE} \bullet \text{READ} \bullet \cdot \mathbf{Q}(s \parallel (np \setminus s)) \cdot \vdash \cdot s}{\text{SOMEONE} \bullet \text{READ} \bullet \cdot \mathbf{Q}(s \parallel (np \setminus s)) \cdot \vdash \cdot s} qL \\
\frac{\text{SOMEONE} \bullet \text{READ} \bullet \cdot \mathbf{Q}(s \parallel (np \setminus s)) \cdot \vdash \cdot s}{\mathbf{Q}(s \parallel (np \setminus s)) \vdash \text{READ} \setminus (\text{SOMEONE} \setminus \cdot s)} dp \\
\frac{\text{BOOK} \vdash \cdot \boxed{n} AxR}{\mathbf{Q}(s \parallel (np \setminus s)) \vdash \text{READ} \setminus (\text{SOMEONE} \setminus \cdot s)} /L \\
\frac{\boxed{\text{EVERY}} \vdash (\text{READ} \setminus (\text{SOMEONE} \setminus \cdot s)) / \text{BOOK}}{\text{SOMEONE} \bullet \text{READ} \bullet \text{EVERY} \bullet \text{BOOK} \vdash \cdot s} dp
\end{array}$$

Secondly, we present an analysis of (2), resulting in interpretation (2a)—the only interpretation.

$$\begin{array}{c}
\vdots \\
\frac{\text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \vdash \cdot \boxed{s}}{\langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot \langle \diamond s \rangle} \diamond L \\
\frac{\langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot \langle \diamond s \rangle}{\boxed{np \setminus s} \vdash \cdot np \cdot \setminus \cdot s} /L \\
\frac{\boxed{\text{SAID}} \vdash (\cdot np \cdot \setminus \cdot s) / \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle}{\cdot np \cdot \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot s} dp \\
\frac{\cdot np \cdot \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot s}{\square \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot np \setminus s} qR \\
\frac{\square \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot np \setminus s}{\square \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot \boxed{np \setminus s}} UnfR \\
\frac{\square \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot \boxed{np \setminus s}}{\boxed{s} \vdash \cdot s} AxL \\
\frac{\boxed{s} \vdash \cdot s}{\text{SOMEONE} \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{WROTE} \bullet \text{EVERY} \bullet \text{BOOK} \rangle \vdash \cdot s} qL
\end{array}$$

As a third example, we show that we can analyse ‘a’ as a weak quantifier, using the type np/n . We give an analysis of (3), resulting in the interpretation where the indefinite takes wide scope—(3b). The quantifier ‘a’ takes scope when it is combined with book.

$$\begin{array}{c}
\vdots \\
\frac{\text{EVERYONE} \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{DEDICATED} \bullet \cdot np \cdot \bullet \text{TO} \bullet \text{MARY} \rangle \vdash \cdot s}{\boxed{np} \vdash (\text{DEDICATED} \setminus (\text{KURT} \setminus [(\text{SAID} \setminus (\text{EVERYONE} \setminus \cdot s))])) / (\text{TO} \bullet \text{MARY})} dp \\
\frac{\boxed{np} \vdash (\text{DEDICATED} \setminus (\text{KURT} \setminus [(\text{SAID} \setminus (\text{EVERYONE} \setminus \cdot s))])) / (\text{TO} \bullet \text{MARY})}{\boxed{\text{A}} \vdash ((\text{DEDICATED} \setminus (\text{KURT} \setminus [(\text{SAID} \setminus (\text{EVERYONE} \setminus \cdot s))])) / (\text{TO} \bullet \text{MARY})) / \text{BOOK}} /L \\
\frac{\boxed{\text{A}} \vdash ((\text{DEDICATED} \setminus (\text{KURT} \setminus [(\text{SAID} \setminus (\text{EVERYONE} \setminus \cdot s))])) / (\text{TO} \bullet \text{MARY})) / \text{BOOK}}{\text{EVERYONE} \bullet \text{SAID} \bullet \langle \text{KURT} \bullet \text{DEDICATED} \bullet \text{A} \bullet \text{BOOK} \bullet \text{TO} \bullet \text{MARY} \rangle \vdash \cdot s} dp
\end{array}$$

Lastly, we present analyses of examples (4) and (5). We demonstrate changing result types using the word ‘which’, which we assign the type

$$\mathbf{Q}(((n \setminus n) / (np \setminus s)) \parallel (np \setminus np)).$$

In the second, for parasitic scope, we deviate slightly from Barker’s [3] treatment of parasitic scope. We assign ‘same’ (and ‘different’) the type

$$\mathbf{Q}(s // (\mathbf{Q}(np \setminus s // ((n // n) \setminus np \setminus s)) \setminus s)).$$

What this type does is quantify over an expression twice—once normally, to plant its top-level quantifier, and once parasitically. Using this type, we can obtain the semantics advocated by Kiselyov [10]. The proofs for these two examples can be found in figure 5.

6 Conclusion

We presented an improvement over Barker and Shan’s [4] NL_{CL} for which derivability is decidable, and which has a normal-form for proof search. In addition, it can analyse scope islands, and distinguish between strong and weak quantifiers, shown by the ability to analyse examples (1-5). Of these examples, (1-3) are representative examples of scope islands and strong and weak quantifiers, for which Kiselyov and Shan [11] provides a purely semantic analysis. The remaining examples, (4) and (5), are examples from the work by Barker and Shan [4] which motivated us to start from their syntactic approach.

References

1. Andreoli, J. (2001). Focussing and proof construction. *Ann. Pure Appl. Logic*, 107(1-3):131–163.
2. Barker, C. (2002). Continuations and the nature of quantification. *Natural Language Semantics*, 10(3):211–242.
3. Barker, C. (2007). Parasitic scope. *Linguistics and Philosophy*, 30(4):407–444.
4. Barker, C. and Shan, C. (2014). *Continuations and Natural Language*, volume 53 of *Oxford Studies in Theoretical Linguistics*. Oxford University Press.
5. Bastenhof, A. (2011). Polarized classical non-associative Lambek calculus and formal semantics. In Pogodalla, S. and Prost, J., editors, *Logical Aspects of Computational Linguistics - 6th International Conference, LACL 2011, Montpellier, France, June 29 - July 1, 2011. Proceedings*, volume 6736 of *Lecture Notes in Computer Science*, pages 33–48. Springer.
6. Bastenhof, A. (2013). Categorical symmetry. <http://dspace.library.uu.nl/handle/1874/273870>.
7. Belnap, N. D. (1982). Display logic. *J. Philosophical Logic*, 11(4):375–417.
8. Finger, M. (1998). Computational solutions for structural constraints. In Moortgat, M., editor, *Logical Aspects of Computational Linguistics, Third International Conference, LACL’98, Grenoble, France, December 14-16, 1998, Selected Papers*, volume 2014 of *Lecture Notes in Computer Science*, pages 11–30. Springer.
9. Heim, I. and Kratzer, A. (1998). *Semantics in generative grammar*, volume 13. Blackwell Oxford.
10. Kiselyov, O. (2015). Compositional semantics of *same*, *different*, *total*. In *Proceedings for ESSLLI 2015 Workshop ‘Empirical Advances in Categorical Grammar’*, pages 71–81.

11. Kiselyov, O. and Shan, C.-c. (2014). Continuation hierarchy and quantifier scope. In *Formal Approaches to Semantics and Pragmatics*, pages 105–134. Springer.
12. Lambek, J. (1961). On the calculus of syntactic types. *Structure of language and its mathematical aspects*, 166:C178.
13. Moortgat, M. (1996). In situ binding: A modal analysis. In Dekker, P. and Stokhof, M., editors, *Proceedings of the Tenth Amsterdam Colloquium*, pages 539–549. Institute for Logic, Language and Computation (ILLC).
14. Moortgat, M. and Moot, R. (2011). Proof nets for the Lambek-Grishin calculus. In *Compositional Methods in Physics and Linguistics*, volume abs/1112.6384.
15. Morrill, G. (1994). *Type logical grammar - categorial logic of signs*. Kluwer.
16. Szabolcsi, A. (2000). The syntax of scope. In *The Handbook of Contemporary Syntactic Theory*, pages 607–633. Wiley-Blackwell.

Atom α ::= $s \mid n \mid np \mid \dots$
 Type A, B ::= $\alpha \mid A \setminus B \mid B / A \mid A \setminus\!\!\setminus B \mid B \setminus\!\!\setminus A \mid \diamond A \mid \square A$
 Struct⁻ Δ ::= $\cdot A \mid \Gamma \setminus \Delta \mid \Delta / \Gamma \mid \Gamma \setminus\!\!\setminus \Delta \mid \Delta \setminus\!\!\setminus \Gamma \mid [\Delta]$
 Struct⁺ Γ ::= $\cdot A \mid \Gamma_1 \bullet \Gamma_2 \mid \Gamma_1 \circ \Gamma_2 \mid \mathbf{I} \mid \mathbf{B} \mid \mathbf{C} \mid \langle \Gamma \rangle$
 Context Σ ::= $\square \mid \Sigma \bullet \Gamma \mid \Gamma \bullet \Sigma$

Pol(s) = -, Pol(n) = +, Pol(np) = +, ...

if Pol(α) = - $\left\{ \frac{\boxed{\alpha} \vdash \cdot \alpha \cdot}{\boxed{\alpha}} \text{Ax}^L \mid \frac{\cdot \alpha \cdot \vdash \boxed{\alpha}}{\alpha \cdot} \text{Ax}^R \right\}$ if Pol(α) = +

if Pol(A) = - $\left\{ \frac{\boxed{A} \vdash \Delta}{\cdot A \cdot \vdash \Delta} \text{Foc}^L \mid \frac{\Gamma \vdash \boxed{A}}{\Gamma \vdash \cdot A \cdot} \text{Foc}^R \right\}$ if Pol(A) = +

$$\begin{array}{c}
 \frac{\Gamma \vdash \boxed{A} \quad \boxed{B} \vdash \Delta}{\boxed{A \setminus B} \vdash \Gamma \setminus \Delta} \setminus L \quad \frac{\Gamma \vdash \cdot A \cdot \setminus \cdot B \cdot}{\Gamma \vdash \cdot A \setminus B \cdot} \setminus R \quad \frac{\Gamma \vdash \boxed{A} \quad \boxed{B} \vdash \Delta}{\boxed{B / A} \vdash \Delta / \Gamma} /L \quad \frac{\Gamma \vdash \cdot B \cdot / \cdot A \cdot}{\Gamma \vdash \cdot B / A \cdot} /R \\
 \\
 \frac{\Gamma \vdash \boxed{A} \quad \boxed{B} \vdash \Delta}{\boxed{A \setminus\!\!\setminus B} \vdash \Gamma \setminus\!\!\setminus \Delta} \setminus\!\!\setminus L \quad \frac{\Gamma \vdash \cdot A \cdot \setminus\!\!\setminus \cdot B \cdot}{\Gamma \vdash \cdot A \setminus\!\!\setminus B \cdot} \setminus\!\!\setminus R \quad \frac{\Gamma \vdash \boxed{A} \quad \boxed{B} \vdash \Delta}{\boxed{B \setminus\!\!\setminus A} \vdash \Delta \setminus\!\!\setminus \Gamma} \setminus\!\!\setminus L \quad \frac{\Gamma \vdash \cdot B \cdot \setminus\!\!\setminus \cdot A \cdot}{\Gamma \vdash \cdot B \setminus\!\!\setminus A \cdot} \setminus\!\!\setminus R \\
 \\
 \frac{\Gamma_2 \vdash \Gamma_1 \setminus \Delta}{\Gamma_1 \bullet \Gamma_2 \vdash \Delta} \text{Res}/\bullet \quad \frac{\Gamma_1 \vdash \Delta / \Gamma_2}{\Gamma_1 \bullet \Gamma_2 \vdash \Delta} \text{Res}/\bullet \quad \frac{\Gamma_2 \vdash \Gamma_1 \setminus\!\!\setminus \Delta}{\Gamma_1 \circ \Gamma_2 \vdash \Delta} \text{Res}/\!\!\setminus \circ \quad \frac{\Gamma_1 \vdash \Delta \setminus\!\!\setminus \Gamma_2}{\Gamma_1 \circ \Gamma_2 \vdash \Delta} \text{Res}/\!\!\setminus \circ \\
 \\
 \frac{\Gamma \vdash \Delta}{\Gamma \circ \mathbf{I} \vdash \Delta} \mathbf{I} \quad \frac{\Gamma_1 \bullet (\Gamma_2 \circ \Gamma_3) \vdash \Delta}{\Gamma_2 \circ ((\mathbf{B} \bullet \Gamma_1) \bullet \Gamma_3) \vdash \Delta} \mathbf{B} \quad \frac{(\Gamma_1 \circ \Gamma_2) \bullet \Gamma_3 \vdash \Delta}{\Gamma_1 \circ ((\mathbf{C} \bullet \Gamma_2) \bullet \Gamma_3) \vdash \Delta} \mathbf{C} \\
 \\
 \frac{\langle \cdot A \cdot \rangle \vdash \Delta}{\cdot \diamond A \cdot \vdash \Delta} \diamond L \quad \frac{\Gamma \vdash \boxed{B}}{\langle \Gamma \rangle \vdash \boxed{\diamond B}} \diamond R \quad \frac{\boxed{A} \vdash \Delta}{\boxed{\square A} \vdash [\Delta]} \square L \quad \frac{\Gamma \vdash [\cdot B \cdot]}{\Gamma \vdash \cdot \square B \cdot} \square R \quad \frac{\Gamma \vdash [\Delta]}{\langle \Gamma \rangle \vdash \Delta} \text{Res}/\square \diamond
 \end{array}$$

Fig. 3. NL_{CL} reworked as a focused display calculus.

$s^* = \mathbf{t}, \quad n^* = \mathbf{et}, \quad np^* = \mathbf{e}, \quad \dots$

$$\begin{array}{l}
 \boxed{\alpha}^+ = \begin{cases} \alpha^* & \text{if Pol}(\alpha) = + \\ ((\alpha^*)^R)^R & \text{if Pol}(\alpha) = - \end{cases} \quad \boxed{\alpha}^- = (\alpha^*)^R \\
 [A \setminus B]^+ = ([A]^+ \times [B]^-)^R \quad [A \setminus B]^- = [A]^+ \times [B]^- \\
 [B / A]^+ = ([B]^- \times [A]^+)^R \quad [B / A]^- = [B]^- \times [A]^+ \\
 [\diamond A]^+ = [A]^+ \quad [\diamond A]^- = ([A]^+)^R \\
 [\square A]^+ = ([A]^+)^R \quad [\square A]^- = [A]^+ \\
 [\mathbf{Q}A]^+ = [A]^+ \quad [\mathbf{Q}A]^- = ([A]^+)^R
 \end{array}$$

$$[[\Gamma \vdash \Delta]] = [[\Gamma]] \vdash [[\Delta]] \quad [[\boxed{A} \vdash \Delta]] = [[\Delta]] \vdash [[A]]^- \quad [[\Gamma \vdash \boxed{A}]] = [[\Gamma]] \vdash [[A]]^+$$

Fig. 4. CPS semantics for focused NL_{CL}.

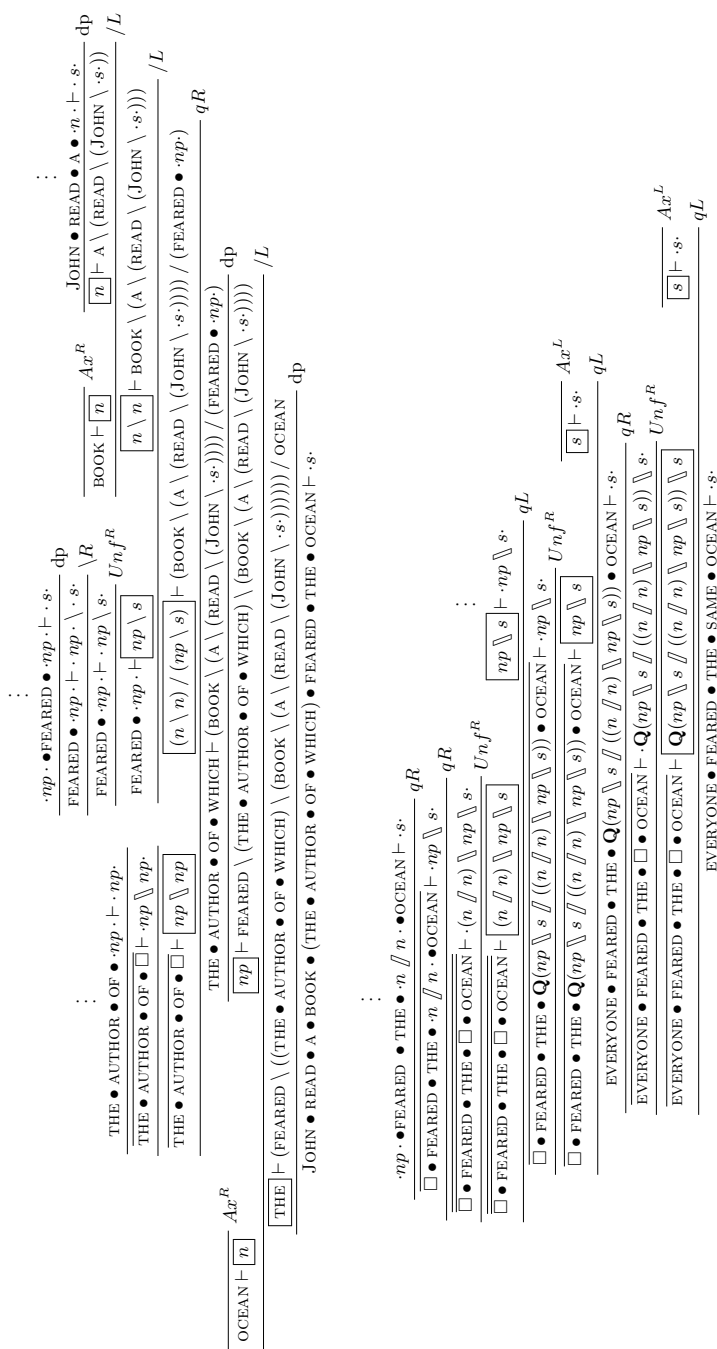


Fig. 5. Analysis of examples (4) and (5).