The String-Meaning Relations Definable by Lambek Grammars and Context-Free Grammars

Makoto Kanazawa, Sylvain Salvati

The String-Meaning Relations Definable by Lambek Grammars and Context-Free Grammars

Makoto Kanazawa[1] and Sylvain Salvati[2]

[1] National Institute of Informatics, 2–1–2 Hitotsubashi, Chiyoda-ku, Tokyo, 101–8430, Japan
[2] INRIA Bordeaux Sud-Ouest, LaBRI, 351, cours de la Libération, F-33405 Talence cedex, France

Abstract. We show that the class of string-meaning relations definable by the following two types of grammars coincides: (i) Lambek grammars where each lexical item is assigned a (suitably typed) lambda term as a representation of its meaning, and the meaning of a sentence is computed according to the lambda-term corresponding to its derivation; and (ii) cycle-free context-free grammars that do not generate the empty string where each rule is associated with a (suitably typed) lambda term that specifies how the meaning of a phrase is determined by the meanings of its immediate constituents.

1 Introduction

It is well known since Pentus's work [4,5,6] that Lambek grammars and context-free grammars can generate the same class of string languages (modulo the empty string). We show that the equivalence continues to hold when semantics is taken into account. Specifically, when Lambek grammars and cycle-free (i.e., finitely ambiguous) context-free grammars are enriched with Montague semantics, they define the same class of relations between (non-empty) strings and meanings (represented as typed λ-terms).

2 Preliminaries

2.1 Lambda Terms over a Higher-Order Signature

If $\mathcal{A}$ is a finite set, then the set $\mathrm{Tp}(\mathcal{A}, \to)$ of *simple types* over $\mathcal{A}$ is the smallest superset of $\mathcal{A}$ such that $A, B \in \mathrm{Tp}(\mathcal{A}, \to)$ implies $A \to B \in \mathrm{Tp}(\mathcal{A}, \to)$. A *higher-order signature* is a triple $\Sigma = (\mathcal{A}, C, \tau)$, where $\mathcal{A}$ is a finite set of *atomic types*, C is a finite set of *constants*, and τ is a function from C to $\mathrm{Tp}(\mathcal{A}, \to)$. If Var is a countably infinite set of *variables*, disjoint from C, then the set $\Lambda(\Sigma)$ of *λ-terms* over Σ is the smallest superset of $C \cup \mathrm{Var}$ such that $M, N \in \Lambda(\Sigma)$ and $x \in \mathrm{Var}$ imply $MN \in \Lambda(\Sigma)$ and $\lambda x.M \in \Lambda(\Sigma)$. A *type environment* is a finite partial function from Var to $\mathrm{Tp}(\mathcal{A}, \to)$, written as a list of typing declarations $x_1 : A_1, \ldots, x_n : A_n$. A λ-term $M[x_1, \ldots, x_n]$ with free variables $x_1, \ldots, x_n$ may be assigned a type B under a typing environment $x_1 : A_1, \ldots, x_n : A_n$, or in symbols, $x_1 : A_1, \ldots, x_n : A_n \vdash_\Sigma M[x_1, \ldots, x_n] : B$. (The subscript Σ may be omitted when $M[x_1, \ldots, x_n]$ is a *pure* λ-term, i.e., does not contain any constants.) Such a *typing judgment* is derived according to the following rules:

$$x : A \vdash_\Sigma x : A \qquad \vdash_\Sigma c : \tau(c)$$

$$\frac{\Gamma \vdash_\Sigma M : A \to B \quad \Delta \vdash_\Sigma N : A}{\Gamma \cup \Delta \vdash_\Sigma MN : B} \qquad \frac{\Gamma \vdash_\Sigma M : B}{\Gamma - \{x : A\} \vdash_\Sigma \lambda x.M : A \to B}$$

(In the last rule, x may not be in the domain of $\Gamma - \{x : A\}$.)

We assume that the reader is familiar with basic notions in λ-calculus, such as β-reduction and β-normal form. We write $M \twoheadrightarrow_\beta M'$ when M β-reduces to M', and write $|M|_\beta$ for the β-normal form of M. As is customary, we adopt the informal practice of identifying λ-terms that are identical modulo renaming of bound variables.

2.2 Product-Free Lambek Calculus

We mostly follow the notations of Pentus [5]. We let $\mathrm{Pr} = \{p_1, p_2, \dots\}$ be a countably infinite set of *primitive types*. If $\mathcal{B}$ is a subset of Pr, we let $\mathrm{Tp}(\mathcal{B}, \backslash, /)$ denote the smallest superset of $\mathcal{B}$ such that $A, B \in \mathrm{Tp}(\mathcal{B}, \backslash, /)$ implies $A\backslash B, B/A \in \mathrm{Tp}(\mathcal{B}, \backslash, /)$. Elements of $\mathrm{Tp}(\mathrm{Pr}, \backslash, /)$ are called *(directional) types*. We let p range over Pr and $A, B, C, \dots$ range over $\mathrm{Tp}(\mathrm{Pr}, \backslash, /)$. When Γ is a finite string of types, we let $|\Gamma|$ denote the number of types in Γ; thus, $|A_1 \dots A_n| = n$.

An expression of the form $\Gamma \to A$, where Γ is a non-empty finite string of types and A is a type, is called a *sequent*. The sequent calculus presentation of the *Lambek calculus* consists of the following axioms and rules:

- Axioms: $p \to p$
- Rules:

$$\frac{\Pi \to A \quad \Gamma B \Delta \to C}{\Gamma \Pi (A\backslash B) \Delta \to C}(\backslash\to) \qquad \frac{A\Pi \to B}{\Pi \to A\backslash B}(\to\backslash) \text{ where } \Pi \neq \varepsilon$$

$$\frac{\Pi \to A \quad \Gamma B \Delta \to C}{\Gamma (B/A) \Pi \Delta \to C}(/\to) \qquad \frac{\Pi A \to B}{\Pi \to B/A}(\to/) \text{ where } \Pi \neq \varepsilon$$

$$\frac{\Pi \to C \quad \Gamma C \Delta \to A}{\Gamma \Pi \Delta \to A}\,\mathrm{Cut}$$

A derivation is *cut-free* if it does not contain any applications of the Cut rule. It is easy to see that every sequent has only finitely many cut-free derivations.

Curry-Howard homomorphism Every derivation $\mathcal{D}$ is associated with a pure λ-term $h(\mathcal{D})$ according to the following rules ($x_1, x_2, \dots$ are specially reserved variables):

- If $\mathcal{D}$ is an axiom $p \to p$, then $h(\mathcal{D}) = x_1$.
- If $\mathcal{D}$ is of the form

$$\frac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Pi \to A\end{array} \quad \begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma B \Delta \to C\end{array}}{\Gamma \Pi (A\backslash B) \Delta \to C}(\backslash\to)$$

  then

$$h(\mathcal{D}) = M[x_1, \dots, x_{i-1}, x_{i+n} N[x_i, \dots, x_{i+n-1}], x_{i+n+1}, \dots, x_{m+n}],$$

  where $|\Gamma| = i - 1$, $h(\mathcal{E}) = M[x_1, \dots, x_m]$, and $h(\mathcal{F}) = N[x_1, \dots, x_n]$.

– If $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{E}\\ A\Pi \to B\end{array}}{\Pi \to A\backslash B}(\to\backslash)$$

then $h(\mathcal{D}) = \lambda z.M[z, x_1, \ldots, x_{m-1}]$, where $h(\mathcal{E}) = M[x_1, \ldots, x_m]$.
– If $\mathcal{D}$ ends in $(/\to)$ or $(\to/)$, $h(\mathcal{D})$ is defined similarly to the preceding two cases.
– If $\mathcal{D}$ is of the form

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Pi \to C\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma C\Delta \to A\end{array}}{\Gamma\Pi\Delta \to A}\,\mathrm{Cut}$$

then

$$h(\mathcal{D}) = M[x_1, \ldots, x_{i-1}, N[x_i, \ldots, x_{i+n-1}], x_{i+n}, \ldots, x_{m+n-1}],$$

where $|\Gamma| = i - 1$, $h(\mathcal{E}) = M[x_1, \ldots, x_m]$, and $h(\mathcal{F}) = N[x_1, \ldots, x_n]$.

We also use h for the mapping from directional types to simple types defined by $h(p) = p$, $h(A\backslash B) = h(B/A) = h(A) \to h(B)$. If $\mathcal{D}$ is a derivation of $A_1 \ldots A_n \to B$, then we always have

$$x_1 : h(A_1), \ldots, x_n : h(A_n) \vdash h(\mathcal{D}) : h(B).$$

Another important fact is that if $\mathcal{D}$ is cut-free, then $h(\mathcal{D})$ is in β-normal form.

Cut elimination

$$\dfrac{p \to p \quad \begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma p\Delta \to A\end{array}}{\Gamma p\Delta \to A}\,\mathrm{Cut} \rightsquigarrow \begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma p\Delta \to A\end{array} \tag{C1}$$

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Pi \to p\end{array}\quad p \to p}{\Pi \to p}\,\mathrm{Cut} \rightsquigarrow \begin{array}{c}\vdots\,\mathcal{F}\\ \Pi \to p\end{array} \tag{C2}$$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\mathcal{F}_1\\ \Pi \to A\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{F}_2\\ \Gamma B\Delta \to C\end{array}}{\Gamma\Pi(A\backslash B)\Delta \to C}(\backslash\to)}{\Gamma\Pi(A\backslash B)\Delta \to C}\quad\begin{array}{c}\vdots\,\mathcal{E}\\ \Phi C\Psi \to D\end{array}}{\Phi\Gamma\Pi(A\backslash B)\Delta\Psi \to D}\,\mathrm{Cut} \rightsquigarrow \dfrac{\begin{array}{c}\vdots\,\mathcal{F}_1\\ \Pi \to A\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{F}_2\\ \Gamma B\Delta \to C\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}\\ \Phi C\Psi \to D\end{array}}{\Phi\Gamma B\Delta\Psi \to D}\,\mathrm{Cut}}{\Phi\Gamma\Pi(A\backslash B)\Delta\Psi \to D}(\backslash\to) \tag{C3}$$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Phi \to C\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1\\ \Pi'C\Pi'' \to A\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}_2\\ \Gamma B\Delta \to D\end{array}}{\Gamma\Pi'C\Pi''(A\backslash B)\Delta \to D}(\backslash\to)}{\Gamma\Pi'\Phi\Pi''(A\backslash B)\Delta \to D}\,\mathrm{Cut} \rightsquigarrow \dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Phi \to C\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1\\ \Pi'C\Pi'' \to A\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}_2\\ \Gamma B\Delta \to D\end{array}}{\Pi'\Phi\Pi'' \to A}\,\mathrm{Cut}}{\Gamma\Pi'\Phi\Pi''(A\backslash B)\Delta \to D}(\backslash\to) \tag{C4}$$

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Phi \to C\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1\\ \Pi \to A\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}_2\\ \Gamma'C\Gamma''B\Delta \to D\end{array}}{\Gamma'C\Gamma''\Pi(A\backslash B)\Delta \to D}(\backslash\to)}{\Gamma'\Phi\Gamma''\Pi(A\backslash B)\Delta \to D}\,\mathrm{Cut} \rightsquigarrow \dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1\\ \Pi \to A\end{array}\quad\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Phi \to C\end{array}\quad\begin{array}{c}\vdots\,\mathcal{E}_2\\ \Gamma'C\Gamma''B\Delta \to D\end{array}}{\Gamma'\Phi\Gamma''B\Delta \to D}\,\mathrm{Cut}}{\Gamma'\Phi\Gamma''\Pi(A\backslash B)\Delta \to D}(\backslash\to) \tag{C5}$$

$$\dfrac{\dfrac{\vdots\,\mathcal{F}}{\Phi \to C} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta' C \Delta'' \to D}(\backslash\!\to)}{\Pi(A\backslash B)\Delta' C \Delta'' \to D}\,\mathrm{Cut}}{\Gamma\Pi(A\backslash B)\Delta'\Phi\Delta'' \to D} \rightsquigarrow \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}}{\Phi \to C} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta' C \Delta'' \to D}\,\mathrm{Cut}}{\Gamma B \Delta' \Phi \Delta'' \to D}(\backslash\!\to)}{\Gamma\Pi(A\backslash B)\Delta'\Phi\Delta'' \to D} \tag{C6}$$

$$\dfrac{\dfrac{\vdots\,\mathcal{F}}{\Phi \to C} \quad \dfrac{\vdots\,\mathcal{E}_1}{A\Pi' C \Pi'' \to B}(\to\backslash)}{\Pi'\Phi\Pi'' \to A\backslash B}\,\mathrm{Cut} \rightsquigarrow \dfrac{\dfrac{\vdots\,\mathcal{F}}{\Phi \to C} \quad \dfrac{\vdots\,\mathcal{E}_1}{A\Pi' C \Pi'' \to B}\,\mathrm{Cut}}{A\Pi'\Phi\Pi'' \to B}(\to\backslash) \tag{C7}$$

Similar to (C3)–(C7), with / in place of \. (C8)–(C12)

$$\dfrac{\dfrac{\vdots\,\mathcal{F}_1}{A\Phi \to B}(\to\backslash)}{\Phi \to A\backslash B} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta \to D}(\backslash\!\to)}{\Gamma\Pi(A\backslash B)\Delta \to D}\,\mathrm{Cut}}{\Gamma\Pi\Phi\Delta \to D} \rightsquigarrow \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\vdots\,\mathcal{F}_1}{A\Phi \to B}\,\mathrm{Cut}}{\Pi\Phi \to B} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta \to D}\,\mathrm{Cut}}{\Gamma\Pi\Phi\Delta \to D} \tag{C13}$$

$$\dfrac{\dfrac{\vdots\,\mathcal{F}_1}{A\Phi \to B}(\to\backslash)}{\Phi \to A\backslash B} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta \to D}(\backslash\!\to)}{\Gamma\Pi(A\backslash B)\Delta \to D}\,\mathrm{Cut}}{\Gamma\Pi\Phi\Delta \to D} \rightsquigarrow \dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Pi \to A} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{A\Phi \to B} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Gamma B \Delta \to D}\,\mathrm{Cut}}{\Gamma A \Phi \Delta \to D}\,\mathrm{Cut}}{\Gamma\Pi\Phi\Delta \to D} \tag{C14}$$

Similar to (C13)–(C14), with / in place of \. (C15)–(C16)

If $\mathcal{D} \rightsquigarrow \mathcal{D}'$ by one of (C1)–(C16), then $h(\mathcal{D}) \twoheadrightarrow_\beta h(\mathcal{D}')$. Every derivation $\mathcal{D}$ reduces to some cut-free derivation $\mathcal{D}'$ by repeated applications of (C1)–(C16).

In general, a derivation may reduce to many different cut-free derivations, although the β-normal λ-terms associated with these derivations are all equal.[3]

2.3 Lambek Grammars with Montague Semantics

A *Lambek grammar with Montague semantics* (*Lambek grammar* for short) is a tuple $G = (\mathcal{B}, \mathcal{T}, \Sigma, f, \mathcal{R}, S)$, where

- $\mathcal{B}$ is a finite subset of Pr,
- $\mathcal{T}$ is a finite set of *terminals*,
- $\Sigma = (\mathcal{A}, C, \tau)$ is a higher-order signature called the *semantic vocabulary*,
- f is a function from $\mathcal{B}$ to $\mathrm{Tp}(\mathcal{A}, \to)$,
- $\mathcal{R}$ is a finite subset of $\mathcal{T} \times \mathrm{Tp}(\mathcal{B}, \backslash, /) \times \Lambda(\Sigma)$ such that if $(a, A, M) \in \mathcal{R}$, then $\vdash_\Sigma M : f(h(A))$,[4]
- S is a distinguished element of $\mathrm{Tp}(\mathcal{B}, \backslash, /)$.

[3] The non-confluence property is due to the fact that (C3) and (C8) have overlapping domains of application with (C4)–(C7) and (C9)–(C12), and the fact that (C13) and (C14) have identical domains of application, as do (C15) and (C16). We note that (C13) and (C15) were not among the rules described by Lambek [3] in his proof of cut elimination. For our purposes, it is convenient, though not essential, to have these rewriting rules, in addition to (C14) and (C16).

[4] Here, f is homomorphically extended to a function from $\mathrm{Tp}(\mathcal{B}, \to)$ to $\mathrm{Tp}(\mathcal{A}, \to)$.

The *string-meaning relation* defined by G is

$$\mathbb{R}(G) = \{\, (a_1 \dots a_n, |M[M_1, \dots, M_n]|_\beta) \mid \mathcal{D} \text{ is a derivation of } B_1 \dots B_n \to S,\ M[x_1, \dots, x_n] = h(\mathcal{D}),\ (a_i, B_i, M_i) \in \mathcal{R} \text{ for } i = 1, \dots, n \,\}.$$

Whenever $(w, M) \in \mathbb{R}(G)$, it holds that $\vdash_\Sigma M : f(h(S))$.

2.4 Context-Free Grammars with Montague Semantics

A *context-free grammar with Montague semantics* (*context-free grammar* for short) is a tuple $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$, where

- $\mathcal{N}$ is a finite set of *nonterminals*,
- $\mathcal{T}$ is a finite set of *terminals*,
- $\Sigma = (\mathcal{A}, C, \tau)$ is a higher-order signature called the *semantic vocabulary*,
- f is a function from $\mathcal{N}$ to $\mathrm{Tp}(\mathcal{A}, \to)$,
- $\mathcal{P}$ is a finite set of *rules* of the form

$$B \to w_0 B_1 w_1 \dots B_n w_n : M[x_1, \dots, x_n] \tag{1}$$

  where $n \geq 0$, $B, B_1, \dots, B_n \in \mathcal{N}$, $w_0, w_1, \dots, w_n \in \mathcal{T}^*$, $M[x_1, \dots, x_n] \in \Lambda(\Sigma)$, and

$$x_1 : f(B_1), \dots, x_n : f(B_n) \vdash_\Sigma M[x_1, \dots, x_n] : f(B),$$

- S is a distinguished element of $\mathcal{N}$ called the *start symbol*.

A *derivation tree of sort* B is a tree of the form $\pi T_1 \dots T_n$, where π is a rule of the form (1) and for $i = 1, \dots, n$, T_i is a derivation tree of sort B_i. We write $\mathbb{D}(G)$ for the set of derivation trees of G (of any sort). The *string yield* of a derivation tree $T = \pi T_1 \dots T_n$ is defined recursively by

$$\mathbf{y}(T) = w_0\, \mathbf{y}(T_1)\, w_1\ \dots\ \mathbf{y}(T_n)\, w_n.$$

The *meaning* of T is defined by

$$\mathbf{m}(T) = M[\mathbf{m}(T_1), \dots, \mathbf{m}(T_n)].$$

Note that whenever T is a derivation tree of sort B, we have

$$\vdash_\Sigma \mathbf{m}(T) : f(B).$$

We write

$$\vdash_G B(w, M)$$

to mean that there is a derivation tree T of sort B such that $\mathbf{y}(T) = w$ and $\mathbf{m}(T) = M$. The *string-meaning relation* defined by G is

$$\mathbb{R}(G) = \{\, (w, |M|_\beta) \mid \vdash_G S(w, M) \,\}.$$

In addition to the notion of a derivation tree, we need the notion of a *derivation tree context*. A derivation tree context is a derivation tree with holes, each denoted by a symbol of the form $\Box_D$, where D is a nonterminal. A *derivation tree context of sort* B is defined inductively as follows:

- $\square_B$ is a derivation tree context of sort B.
- If π is a rule of the form (1) and T_i is a derivation tree context of sort B_i for $i = 1, \ldots, n$, then $\pi T_1 \ldots T_n$ is a derivation tree context of sort B.

The yield and meaning of a derivation tree context are defined as follows:

$$\begin{aligned}
\mathbf{y}(\square_D) &= D,\\
\mathbf{y}(\pi T_1 \ldots T_n) &= w_0\, \mathbf{y}(T_1)\, w_1 \ \ldots\ \mathbf{y}(T_n)\, w_n,\\
\mathbf{m}(\square_D) &= x_1,\\
\mathbf{m}(\pi T_1 \ldots T_n) &= M[P_1[x_1, \ldots, x_{k_1}], \ldots, P_n[x_{k_1+\cdots+k_{n-1}+1}, \ldots, x_{k_1+\cdots+k_n}]],\\
&\qquad \text{where } P_i[x_1, \ldots, x_{k_i}] = \mathbf{m}(T_i).
\end{aligned}$$

If T is a derivation tree context of sort B with n holes, labeled $\square_{D_1}, \ldots, \square_{D_n}$, respectively, from left to right, then

$$\begin{gathered}
\mathbf{y}(T) \in \mathcal{T}^* D_1 \mathcal{T}^* \ldots D_n \mathcal{T}^*,\\
x_1 : f(D_1), \ldots, x_n : f(D_n) \vdash_\Sigma \mathbf{m}(T) : f(B).
\end{gathered}$$

We write

$$\vdash_G B(\gamma, M)$$

to mean that there is a derivation tree context T of sort B such that $\mathbf{y}(T) = \gamma$ and $\mathbf{m}(T) = M$.

Let T be a derivation tree context of sort B with m holes, and $i \in \{1, \ldots, m\}$. If $\square_D$ is the label of the i-th hole (from the left) of T and U is a derivation tree context of sort D with n holes, then the result of replacing the i-th hole of T by U, call it T', is a derivation tree context of sort B with $m + n - 1$ holes. If $\gamma D \delta = \mathbf{y}(T)$, where $\gamma \in (\mathcal{T}^* \mathcal{N})^{i-1} \mathcal{T}^*$, then

$$\mathbf{y}(T') = \gamma \mathbf{y}(U) \delta,$$

and

$$\mathbf{m}(T') = M[x_1, \ldots, x_{i-1}, N[x_i, \ldots, x_{i+n-1}], x_{i+n}, \ldots, x_{m+n-1}],$$

where $M[x_1, \ldots, x_m] = \mathbf{m}(T)$ and $N[x_1, \ldots, x_n] = \mathbf{m}(U)$.

If we ignore the components Σ, f of $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ and remove colons and λ-terms from the rules in $\mathcal{P}$, we get an ordinary context-free grammar. We write $\Rightarrow_G$ for the relation of one-step rewriting associated with this context-free grammar. We write $\Rightarrow_G^+$ and $\Rightarrow_G^*$ for the transitive and reflexive transitive closure of this relation, respectively. Clearly, for every $B \in \mathcal{N}$, $w \in \mathcal{T}^*$, and $\delta \in (\mathcal{N} \cup \mathcal{T})^*$, we have

- $B \Rightarrow_G^* w$ iff there is a derivation tree T of sort B such that $\mathbf{y}(T) = w$, and
- $B \Rightarrow_G^* \delta$ iff there is a derivation tree context T of sort B such that $\mathbf{y}(T) = \delta$.

We write $\mathbb{L}(G)$ for

$$\{\, w \in \mathcal{T}^* \mid S \Rightarrow_G^* w \,\} = \{\, \mathbf{y}(T) \mid T \text{ is a derivation tree of } G \text{ of sort } S \,\}.$$

We call $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ *cycle-free* if G does not allow a cycle $B \Rightarrow_G^+ B$ for any $B \in \mathcal{N}$. If G is cycle-free, then for any $w \in \mathcal{T}^*$, the set $\{\, T \in \mathbb{D}(G) \mid \mathbf{y}(T) = w \,\}$ is finite, and a fortiori, the set of meanings associated with each w, $\{\, M \mid (w, M) \in \mathbb{R}(G) \,\}$, is finite.

3 From Lambek to Context-Free Grammars

3.1 Pentus's Interpolation Lemma and Cut Elimination

Pentus's proof of his interpolation lemma for product-free Lambek calculus (Lemma 7 of [5]) amounts to an algorithm that, given a cut-free derivation $\mathcal{D}$ of $\Gamma \to C$ and a partition (Φ, Θ, Ψ) of Γ (i.e., $\Phi\Theta\Psi = \Gamma$), returns a sequence of cut-free derivations $(\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_n)$ $(n \geq 0)$ satisfying the following properties:

(i) for $i = 1, \ldots, n$, $\mathcal{D}_i$ is a derivation of $\Theta_i \to D_i$,
(ii) $\Theta_1 \ldots \Theta_n = \Theta$,
(iii) $\mathcal{D}_0$ is a derivation of $\Phi D_1 \ldots D_n \Psi \to C$,
(iv) for every atomic type p, if p occurs in D_i, then p occurs in both Θ_i and $\Phi\Psi C$.

We may add the following condition:

(v)
$$\dfrac{\dfrac{\vdots\,\mathcal{D}_1}{\Theta_1 \to D_1} \qquad \dfrac{\dfrac{\vdots\,\mathcal{D}_n}{\Theta_n \to D_n} \quad \dfrac{\vdots\,\mathcal{D}_0}{\Phi D_1 \ldots D_n \Psi \to C}\,\text{Cut}}{\Phi D_1 \ldots D_{n-1} \Theta_n \Psi \to C}\,\text{Cut}}{\Phi\Theta_1 \ldots \Theta_n \Psi \to C}\,\text{Cut} \quad \rightsquigarrow^* \quad \dfrac{\vdots\,\mathcal{D}}{\Phi\Theta_1 \ldots \Theta_n \Psi \to C}$$

That is, the cut-free derivations found by Pentus's interpolation algorithm can be combined by the Cut rule to form a derivation that reduces to the original one.[5]

Lemma 1. *Condition* (v) *holds of Pentus's algorithm for interpolation.*

Proof (sketch). We refer to the numbering of cases used in Pentus's proof [5]. Square brackets indicate the selected (i.e., middle) part of the three-way partition of antecedents. We only treat two subcases of Case 4.

Case 4. $\mathcal{D}$ ends in an application of $(\backslash\to)$.

Case 4e.
$$\mathcal{D} = \dfrac{\dfrac{\vdots\,\mathcal{F}}{\Pi \to A} \quad \dfrac{\vdots\,\mathcal{E}}{\Gamma'[\Gamma'' B \Delta']\Delta'' \to C}}{\Gamma'[\Gamma''\Pi(A\backslash B)\Delta']\Delta'' \to C}\,(\backslash\to)$$

By induction hypothesis, we have

$$\dfrac{\dfrac{\vdots\,\mathcal{E}_1}{\Theta_1 \to E_1} \qquad \dfrac{\dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma' E_1 \ldots E_r \Delta'' \to C}\,\text{Cut}}{\Gamma' E_1 \ldots E_{r-1} \Theta_r \Delta'' \to C}\,\text{Cut}}{\Gamma'\Theta_1 \ldots \Theta_r \Delta'' \to C}\,\text{Cut} \quad \rightsquigarrow^* \quad \dfrac{\vdots\,\mathcal{E}}{\Gamma'\Theta_1 \ldots \Theta_r \Delta'' \to C}$$

[5] For the purpose of the present paper, it is actually enough to know that the λ-terms corresponding to the two derivations in (v) are β-equal, but the stronger property may be of independent interest. For an analogous (but more involved) property of interpolation in the sequent calculus for intuitionistic implicational logic, see [1].

where $\Theta_1 \dots \Theta_r = \Gamma'' B \Delta'$. Let k, Ξ, Υ be such that

$$\Theta_1 \dots \Theta_{k-1} \Xi = \Gamma'', \qquad \Theta_k = \Xi B \Upsilon, \qquad \Upsilon \Theta_{k+1} \dots \Theta_r = \Delta'.$$

In this case, Pentus's algorithm gives $(\mathcal{E}_0, \mathcal{E}_1, \dots, \mathcal{E}_{k-1}, \tilde{\mathcal{E}}_k, \mathcal{E}_{k+1}, \dots, \mathcal{E}_r)$, where

$$\tilde{\mathcal{E}}_k = \dfrac{\begin{array}{cc}\vdots\,\mathcal{F} & \vdots\,\mathcal{E}_k \\ \Pi \to A & \Xi B \Upsilon \to E_k\end{array}}{\Xi \Pi (A \backslash B) \Upsilon \to E_k}\,(\backslash\!\to)$$

We have

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1 \\ \Theta_1 \to E_1\end{array} \quad \dfrac{\begin{array}{c}\vdots\,\mathcal{E}_{k-1} \\ \Theta_{k-1} \to E_{k-1}\end{array} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{F} & \vdots\,\mathcal{E}_k \\ \Pi \to A & \Xi B \Upsilon \to E_k\end{array}}{\Xi \Pi (A \backslash B) \Upsilon \to E_k}\,(\backslash\!\to) \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_{k+1} & \\ \Theta_{k+1} \to E_{k+1} & \Gamma' E_1 \dots E_{k+1} \Theta_{k+2} \dots \Theta_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_k \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' E_1 \dots E_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' E_1 \dots E_{k-2} \Theta_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_r & \vdots\,\mathcal{E}_0 \\ \Theta_r \to E_r & \Gamma' E_1 \dots E_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_{r-1} \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' \Theta_1 \dots \Theta_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}$$

$$\rightsquigarrow \text{(C3)} \quad \dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1 \\ \Theta_1 \to E_1\end{array} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_{k-1} & \vdots\,\mathcal{F} \\ \Theta_{k-1} \to E_{k-1} & \Pi \to A\end{array} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_k & \\ \Xi B \Upsilon \to E_k & \Gamma' E_1 \dots E_k \Theta_{k+1} \dots \Theta_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_{k-1} \Xi B \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' E_1 \dots E_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,(\backslash\!\to)}{\Gamma' E_1 \dots E_{k-2} \Theta_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_{k+1} & \\ \Theta_{k+1} \to E_{k+1} & \Gamma' E_1 \dots E_{k+1} \Theta_{k+2} \dots \Theta_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_k \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_r & \vdots\,\mathcal{E}_0 \\ \Theta_r \to E_r & \Gamma' E_1 \dots E_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_{r-1} \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' \Theta_1 \dots \Theta_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}$$

$$\rightsquigarrow^{*} \text{(C5)} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{F} & \dfrac{\begin{array}{c}\vdots\,\mathcal{E}_1 \\ \Theta_1 \to E_1\end{array} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_{k-1} & \\ \Xi B \Upsilon \to E_k & \Gamma' E_1 \dots E_{k-1} \Xi B \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_{k-1} \Xi B \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' E_1 \dots E_{k-2} \Theta_{k-1} \Xi B \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_{k+1} & \\ \Theta_{k+1} \to E_{k+1} & \Gamma' E_1 \dots E_{k+1} \Theta_{k+2} \dots \Theta_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_k \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_r & \vdots\,\mathcal{E}_0 \\ \Theta_r \to E_r & \Gamma' E_1 \dots E_r \Delta'' \to C\end{array}}{\Gamma' E_1 \dots E_{r-1} \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' \Theta_1 \dots \Theta_{k-1} \Xi B \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,\mathrm{Cut}}{\Gamma' \Theta_1 \dots \Theta_{k-1} \Xi \Pi (A \backslash B) \Upsilon \Theta_{k+1} \dots \Theta_r \Delta'' \to C}\,(\backslash\!\to)$$

$$\rightsquigarrow^* \text{ by I.H.} \quad \dfrac{\begin{array}{cc}\vdots\,\mathcal{F} & \vdots\,\mathcal{E}\\ \Pi \to A & \Gamma'\Theta_1\ldots\Theta_{k-1}\Xi B\Upsilon\Theta_{k+1}\ldots\Theta_r\Delta'' \to C\end{array}}{\Gamma'\Theta_1\ldots\Theta_{k-1}\Xi\Pi(A\backslash B)\Upsilon\Theta_{k+1}\ldots\Theta_r\Delta'' \to C}(\backslash\to)$$

Case 4f.

$$\mathcal{D} = \dfrac{\begin{array}{cc}\vdots\,\mathcal{F} & \vdots\,\mathcal{E}\\ [\Pi']\Pi'' \to A & \Gamma[B\Delta']\Delta'' \to C\end{array}}{\Gamma\Pi'[\Pi''(A\backslash B)\Delta']\Delta'' \to C}(\backslash\to)$$

where $\Pi' \neq \varepsilon$. By induction hypothesis, we have

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,\mathcal{F}_1 & \\ \Xi_1 \to F_1 & F_1\Xi_2\ldots\Xi_m\Pi'' \to A\end{array}}{\Xi_1\ldots\Xi_m\Pi'' \to A}\text{Cut} \qquad \begin{array}{cc}\vdots\,\mathcal{F}_m & \vdots\,\mathcal{F}_0\\ \Xi_m \to F_m & F_1\ldots F_m\Pi'' \to A\end{array}}{F_1\ldots F_{m-1}\Xi_m\Pi'' \to A}\text{Cut} \rightsquigarrow^* \begin{array}{c}\vdots\,\mathcal{F}\\ \Xi_1\ldots\Xi_m\Pi'' \to A\end{array}$$

$$\dfrac{\dfrac{\begin{array}{cc}\vdots\,\mathcal{E}_1 & \\ \Theta_1 \to E_1 & \Gamma E_1\Theta_2\ldots\Theta_r\Delta'' \to C\end{array}}{\Gamma\Theta_1\ldots\Theta_r\Delta'' \to C}\text{Cut} \qquad \begin{array}{cc}\vdots\,\mathcal{E}_r & \vdots\,\mathcal{E}_0\\ \Theta_r \to E_r & \Gamma E_1\ldots E_r\Delta'' \to C\end{array}}{\Gamma E_1\ldots E_{r-1}\Theta_r\Delta'' \to C}\text{Cut} \rightsquigarrow^* \begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma\Theta_1\ldots\Theta_r\Delta'' \to C\end{array}$$

where $m, r \geq 1$, $\Xi_1\ldots\Xi_m = \Pi'$, and $\Theta_1\ldots\Theta_r = B\Delta'$. In this case, Pentus's algorithm gives $(\tilde{\mathcal{E}}_0, \tilde{\mathcal{E}}_1, \mathcal{E}_2, \ldots, \mathcal{E}_r)$, where

$$\tilde{\mathcal{E}}_0 = \dfrac{\begin{array}{cc}\vdots\,\mathcal{F}_m & \\ \Xi_m \to F_m & \Gamma\Xi_1\ldots\Xi_{m-1}(F_{m-1}\backslash(\ldots\backslash(F_1\backslash E_1)\ldots))E_2\ldots E_r\Delta'' \to C\end{array}}{\Gamma\Xi_1\ldots\Xi_m(F_m\backslash(\ldots\backslash(F_1\backslash E_1)\ldots))E_2\ldots E_r\Delta'' \to C}(\backslash\to)$$

$$\tilde{\mathcal{E}}_1 = \dfrac{\dfrac{\dfrac{\begin{array}{cc}\vdots\,\mathcal{F}_0 & \vdots\,\mathcal{E}_1\\ F_1\ldots F_m\Pi'' \to A & B\Upsilon \to E_1\end{array}}{F_1\ldots F_m\Pi''(A\backslash B)\Upsilon \to E_1}(\backslash\to)}{F_2\ldots F_m\Pi''(A\backslash B)\Upsilon \to (F_1\backslash E_1)}(\to\backslash)}{\Pi''(A\backslash B)\Upsilon \to (F_m\backslash(\ldots\backslash(F_1\backslash E_1)\ldots))}(\to\backslash)$$

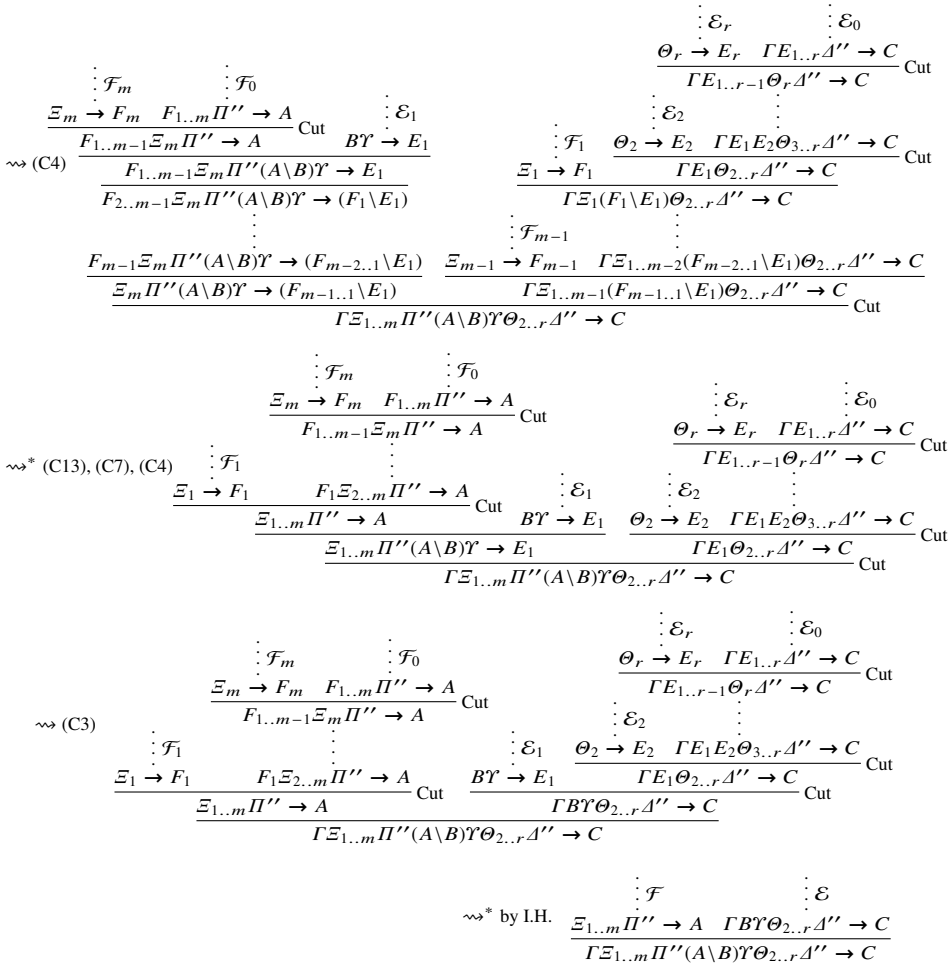
with $\Theta_1 = B\Upsilon$ and $\Delta' = \Upsilon\Theta_2\ldots\Theta_r$. In the following derivations, we abbreviate a sequence of types $C_i\ldots C_j$ by $C_{i..j}$, a concatenation of sequences of types $\Gamma_i\ldots\Gamma_j$ by $\Gamma_{i..j}$, and a type of the form $(C_i\backslash(\ldots\backslash(C_j\backslash D)\ldots))$ by $(C_{i..j}\backslash D)$. We also omit rule labels other than "Cut". We have

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_0}{F_{1..m}\Pi'' \to A} \quad \dfrac{\vdots\,\mathcal{E}_1}{B\Upsilon \to E_1}}{F_{1..m}\Pi''(A\backslash B)\Upsilon \to E_1}}{F_{2..m}\Pi''(A\backslash B)\Upsilon \to (F_1\backslash E_1)} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma\Xi_1(F_1\backslash E_1)E_{2..r}\Delta'' \to C}}{\Gamma\Xi_{1..m}(F_{m..1}\backslash E_1)E_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}(F_{m..1}\backslash E_1)E_2\Theta_{3..r}\Delta'' \to C}\,\text{Cut}}{F_m\Pi''(A\backslash B)\Upsilon \to (F_{m-1..1}\backslash E_1)} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma\Xi_1(F_1\backslash E_1)E_{2..r}\Delta'' \to C}}{\Gamma E_{1..r}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Gamma E_1\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}(F_{m..1}\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Pi''(A\backslash B)\Upsilon \to (F_{m..1}\backslash E_1)} \quad \dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma\Xi_1(F_1\backslash E_1)E_{2..r}\Delta'' \to C}}{\Gamma E_{1..r}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Gamma E_1\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}(F_{m..1}\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}\Pi''(A\backslash B)\Upsilon\Theta_{2..r}\Delta'' \to C}\,\text{Cut}$$

$$\rightsquigarrow^*\ (\text{C6})\quad \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_0}{F_{1..m}\Pi'' \to A} \quad \dfrac{\vdots\,\mathcal{E}_1}{B\Upsilon \to E_1}}{F_{1..m}\Pi''(A\backslash B)\Upsilon \to E_1}}{F_{2..m}\Pi''(A\backslash B)\Upsilon \to (F_1\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_1\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_1(F_1\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{F_m\Pi''(A\backslash B)\Upsilon \to (F_{m-1..1}\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_{1..r-1}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Pi''(A\backslash B)\Upsilon \to (F_{m..1}\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_{1..r-1}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}(F_{m..1}\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}\Pi''(A\backslash B)\Upsilon\Theta_{2..r}\Delta'' \to C}$$

$$\rightsquigarrow\ (\text{C13})\quad \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_0}{F_{1..m}\Pi'' \to A} \quad \dfrac{\vdots\,\mathcal{E}_1}{B\Upsilon \to E_1}}{F_{1..m}\Pi''(A\backslash B)\Upsilon \to E_1}}{F_{2..m}\Pi''(A\backslash B)\Upsilon \to (F_1\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{F_{m-1}F_m\Pi''(A\backslash B)\Upsilon \to (F_{m-2..1}\backslash E_1)}{F_m\Pi''(A\backslash B)\Upsilon \to (F_{m-1..1}\backslash E_1)}\,\text{Cut}}{\Xi_m\Pi''(A\backslash B)\Upsilon \to (F_{m-1..1}\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_{m-1}}{\Xi_{m-1} \to F_{m-1}} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_{1..r-1}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m-1}(F_{m-1..1}\backslash E_1)\Theta_{2..r}\Delta'' \to C} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_1\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_1(F_1\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}\Pi''(A\backslash B)\Upsilon\Theta_{2..r}\Delta'' \to C}\,\text{Cut}$$

$$\rightsquigarrow^*\ (\text{C7})\quad \dfrac{\dfrac{\dfrac{\dfrac{\vdots\,\mathcal{F}_0}{F_{1..m}\Pi'' \to A} \quad \dfrac{\vdots\,\mathcal{E}_1}{B\Upsilon \to E_1}}{F_{1..m}\Pi''(A\backslash B)\Upsilon \to E_1}}{F_{2..m-1}\Xi_m\Pi''(A\backslash B)\Upsilon \to (F_1\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_m}{\Xi_m \to F_m} \quad \dfrac{F_{1..m}\Pi'' \to A \quad B\Upsilon \to E_1}{F_{1..m}\Pi''(A\backslash B)\Upsilon \to E_1}}{F_{1..m-1}\Xi_m\Pi''(A\backslash B)\Upsilon \to E_1}\,\text{Cut}}{F_{m-1}\Xi_m\Pi''(A\backslash B)\Upsilon \to (F_{m-2..1}\backslash E_1)} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_{m-1}}{\Xi_{m-1} \to F_{m-1}} \quad \dfrac{\vdots\,\mathcal{E}_r}{\Theta_r \to E_r} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_{1..r-1}\Theta_r\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m-1}(F_{m-1..1}\backslash E_1)\Theta_{2..r}\Delta'' \to C} \quad \dfrac{\dfrac{\vdots\,\mathcal{E}_2}{\Theta_2 \to E_2} \quad \dfrac{\dfrac{\vdots\,\mathcal{F}_1}{\Xi_1 \to F_1} \quad \dfrac{\vdots\,\mathcal{E}_0}{\Gamma E_{1..r}\Delta'' \to C}}{\Gamma E_1\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_1(F_1\backslash E_1)\Theta_{2..r}\Delta'' \to C}\,\text{Cut}}{\Gamma\Xi_{1..m}\Pi''(A\backslash B)\Upsilon\Theta_{2..r}\Delta'' \to C}$$



The remaining cases are handled similarly. □

3.2 Pentus's Construction

Define

$$\|p\| = 1, \qquad \|A\backslash B\| = \|A\| + \|B\|, \qquad \|B/A\| = \|B\| + \|A\|,$$
$$\|A_1 \dots A_n\| = \|A_1\| + \dots + \|A_n\|.$$

An (m,q)-*type* is a type A such that $\|A\| \le m$ and the atomic types that occur in A are among $p_1, \dots, p_q$. A sequent $A_1 \dots A_n \to C$ is an (m,q)-*sequent* if $A_1, \dots, A_n, C$ are all (m,q)-types. The class of Lcut(m,q)-*derivations* are defined inductively as follows:

- A cut-free derivation of $A_1 \dots A_n \to C$ is an Lcut(m,q)-derivation if $A_1 \dots A_n \to C$ is an (m,q)-sequent and $\|A_1 \dots A_n\| \le 2m$.

– If $\mathcal{F}$ is an Lcut(m,q)-derivation of $\Pi \to C$ and $\mathcal{E}$ is an Lcut(m,q)-derivation of $\Gamma C \Delta \to A$, then

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Pi \to C\end{array} \qquad \begin{array}{c}\vdots\,\mathcal{E}\\ \Gamma C \Delta \to A\end{array}}{\Gamma \Pi \Delta \to A}\,\mathrm{Cut}$$

is an Lcut(m,q)-derivation.

Pentus uses his interpolation lemma to prove that every derivable (m,q)-sequent has an Lcut(m,q)-derivation (Theorem 1 of [5]). With Lemma 1, we can strengthen this theorem to the following:

Lemma 2. *For every cut-free derivation $\mathcal{D}$ of an (m,q)-sequent, there is an* Lcut(m,q)*-derivation $\mathcal{D}'$ of the same sequent such that $\mathcal{D}' \rightsquigarrow^* \mathcal{D}$.*

Let $G = (\mathcal{B}, \mathcal{T}, \Sigma, f, \mathcal{R}, S)$ be a Lambek grammar with Montague semantics. Let q be the least number such that $\mathcal{B} \subseteq \{p_1, \dots, p_q\}$ and let $m = \max(\{\, \|B\| \mid (a, B, M) \in \mathcal{R} \,\} \cup \{\|S\|\})$. Construct a context-free grammar with Montague semantics $G_{\mathrm{cf}} = (\mathcal{N}, \mathcal{T}, \Sigma, f', \mathcal{P}, S)$, where

$$\begin{aligned}
\mathcal{N} &= \{\, B \in \mathrm{Tp}(\mathcal{B}, \backslash, /) \mid B \text{ is an } (m,q)\text{-type} \,\},\\
f'(B) &= f(h(B)) \quad \text{for all } B \in \mathcal{N},\\
\mathcal{P} &= \{\, C \to A_1 \dots A_n : h(\mathcal{D}) \mid C, A_1, \dots, A_n \in \mathcal{N}, \|A_1 \dots A_n\| \le 2m,\\
&\qquad\qquad \mathcal{D} \text{ is a cut-free derivation of } A_1 \dots A_n \to C \,\} \cup\\
&\quad \{\, B \to a : M \mid (a, B, M) \in \mathcal{R} \,\}.
\end{aligned}$$

Note that $\mathcal{N}$ and $\mathcal{P}$ are both finite.

Lemma 3. *Let $B_1, \dots, B_n, C \in \mathcal{N}$.*

(i) *If $\mathcal{D}$ is an* Lcut(m,q)*-derivation of $B_1 \dots B_n \to C$, then there is a derivation tree context T of G_{cf} of sort C such that $\mathbf{y}(T) = B_1 \dots B_n$ and $\mathbf{m}(T) = h(\mathcal{D})$.*

(ii) *If T is a derivation tree context of G_{cf} of sort C such that $\mathbf{y}(T) = B_1 \dots B_n$, then there is an* Lcut$(m,q)$*-derivation $\mathcal{D}$ of $B_1 \dots B_n \to C$ such that $h(\mathcal{D}) = \mathbf{m}(T)$.*

Theorem 4. *For any Lambek grammar with Montague semantics G, $\mathbb{R}(G) = \mathbb{R}(G_{\mathrm{cf}})$.*

The grammar G_{cf} contains cycles $B \Rightarrow^+_{G_{\mathrm{cf}}} B$. The next lemma allows us to modify the construction to obtain a grammar G'_{cf} that has no rule of the form $B \to A : M[x_1]$.

Lemma 5. *For any* Lcut(m,q)*-derivation $\mathcal{D}$, there is an* Lcut(m,q)*-derivation $\mathcal{D}'$ of the same sequent such that $|h(\mathcal{D})|_\beta = |h(\mathcal{D}')|_\beta$ and no sequent of the form $A \to B$ appears in $\mathcal{D}'$ as a right premise of the Cut rule.*

Proof (sketch). Use the following rewriting to transform $\mathcal{D}$ into $\mathcal{D}'$.

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\mathcal{F}_1\\ \Pi \to C\end{array} \qquad \begin{array}{c}\vdots\,\mathcal{F}_2\\ \Gamma C \Delta \to A\end{array}}{\Gamma \Pi \Delta \to A}\,\mathrm{Cut} \qquad \begin{array}{c}\vdots\,\mathcal{E}\\ A \to B\end{array}}{\Gamma \Pi \Delta \to B}\,\mathrm{Cut} \quad \dashrightarrow \quad \dfrac{\begin{array}{c}\vdots\,\mathcal{F}_1\\ \Pi \to C\end{array} \qquad \dfrac{\begin{array}{c}\vdots\,\mathcal{F}_2\\ \Gamma C \Delta \to A\end{array} \qquad \begin{array}{c}\vdots\,\mathcal{E}\\ A \to B\end{array}}{\Gamma C \Delta \to B}\,\mathrm{Cut}}{\Gamma \Pi \Delta \to B}\,\mathrm{Cut}$$

$$\dfrac{\begin{array}{c}\vdots\,\mathcal{F}\\ \Gamma \to A\end{array} \qquad \begin{array}{c}\vdots\,\mathcal{E}\\ A \to B\end{array}}{\Gamma \to B}\,\mathrm{Cut} \rightsquigarrow^* \text{a cut-free derivation of } \Gamma \to B \quad \text{where } \|\Gamma\| \le 2m \qquad \square$$

4 From Context-Free to Lambek Grammars

4.1 From Greibach Normal Form Context-Free Grammars to Lambek Grammars

As with the case of context-free grammars without semantics, the conversion from context-free grammars with Montague semantics to Lambek grammars is based on the Greibach normal form. A context-free grammar with Montague semantics $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ is said to be in *Greibach normal form* if the associated grammar without semantics is in Greibach normal form, i.e., if each rule in $\mathcal{P}$ is of the form $B \to aC_1 \dots C_n : M[x_1, \dots, x_n]$, where $a \in \mathcal{T}$ and $C_i \in \mathcal{N}$. Such a grammar can be converted to a Lambek grammar $G' = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{R}, S)$ by letting $\mathcal{R}$ consist of all triples

$$(a, (\dots (B/C_n)/\dots)/C_1, \lambda z_1 \dots z_n.M[z_1, \dots, z_n])$$

such that $B \to aC_1 \dots C_n : M[x_1, \dots, x_n]$ is a rule in $\mathcal{P}$. (Here, we assume that $\mathcal{N}$ is identified with some finite subset of Pr.)

4.2 Greibach Normal Form Transformation of Context-Free Grammars with Montague Semantics

We describe a procedure for converting a cycle-free context-free grammar with Montague semantics G with $\varepsilon \notin \mathbb{L}(G)$ into an equivalent one in Greibach normal form. This is done in five steps. The first step eliminates all ε-rules from the grammar. The second step eliminates all unit rules. The third step performs the left-corner transform, well-known from the work of Rosenkrantz and Lewis [7], but enriched with semantics. The fourth step takes the result of the previous step and converts it into extended Greibach normal form. The last step then converts it into Greibach normal form. The first four steps roughly mirror the procedure presented in the technical report by Kanazawa and Yoshinaka [2].

Suppose that $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ is a cycle-free grammar such that $\varepsilon \notin \mathbb{L}(G)$. Let us call a nonterminal B *nullable* if $B \Rightarrow_G^* \varepsilon$. By assumption, S is not nullable. Note that the binary relation $\Rightarrow_G^+$ restricted to $\mathcal{N}$ is a strict partial order. When $A \Rightarrow_G^+ B$ holds, we consider A "less than" B with respect to this partial order.

Elimination of ε-rules A rule of the form $B \to \varepsilon : M$ is called an ε-rule. Let C be a nullable nonterminal that is maximal with respect to the strict partial order $\Rightarrow_G^+$. Let $\mathcal{P}_0$ be the set of all ε-rules in $\mathcal{P}$ with C as the left-hand side nonterminal. For each rule π of the form

$$B \to w_0 B_1 w_1 \dots B_n w_n : M[x_1, \dots, x_n],$$

let $\pi \circ \mathcal{P}_0$ consist of all rules of the form

$$B \to w_0 \beta_1 w_1 \dots \beta_n w_n : M[Q_1, \dots, Q_n]$$

such that for some $k_1, \dots, k_n$, each $i \in \{1, \dots, n\}$ satisfies either

- $\beta_i = B_i$, $Q_i = x_{k_i}$, and $k_i = k_{i-1} + 1$, or

– $\beta_i = \varepsilon$, $B_i = C$, $\mathcal{P}_0$ contains the rule $C \to \varepsilon : Q_i$, and $k_i = k_{i-1}$,

where $k_0 = 0$. Let

$$\mathcal{P}' = \bigcup_{\pi \in \mathcal{P} - \mathcal{P}_0} \pi \circ \mathcal{P}_0,$$
$$G' = (\mathcal{N}, \mathcal{T}, \Sigma, \mathcal{P}', S).$$

Lemma 6. *For every $B \in \mathcal{N}$ and $w \in \mathcal{T}^+$, the following are equivalent:*

(i) $\vdash_G B(w, N)$.
(ii) *Either* $\vdash_{G'} B(w, N)$ *or* $B = C$, $w = \varepsilon$, *and* $\mathcal{P}_0$ *contains the rule* $C \to \varepsilon : N$.

Lemma 7. *For every $B \in \mathcal{N}$, B is nullable in G' if and only if $B \neq C$ and B is nullable in G.*

Lemma 8. *For every $B, B' \in \mathcal{N}$, $B \Rightarrow^+_{G'} B'$ if and only if $B \Rightarrow^+_G B'$.*

By Lemma 6, $\mathbb{R}(G') = \mathbb{R}(G)$, and by Lemmas 7 and 8, G' is a cycle-free grammar with one fewer nullable nonterminals than G. By repeating this procedure, we can turn G into an equivalent one that is cycle-free and contains no ε-rules.

Elimination of unit rules A *unit rule* is a rule of the form $B \to B_1 : M[x_1]$. If $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ is a cycle-free grammar with no ε-rules, we can eliminate unit rules from G by a procedure similar to the one used for the previous step. Let C be a nonterminal in $\mathcal{N}$ that is maximal, but not minimal, with respect to the strict partial order $\Rightarrow^+_G$. This means that there is a unit rule with C as its right-hand side nonterminal, but there is no unit rule with C as its left-hand side nonterminal. Let $\mathcal{P}_{\text{left}}$ be the set of all rules in $\mathcal{P}$ with C as their left-hand side nonterminal, and let $\mathcal{P}_{\text{right}}$ be the set of all unit rules in $\mathcal{P}$ with C as their *right-hand* side nonterminal. Let $\mathcal{P}_{\text{right}} \circ \mathcal{P}_{\text{left}}$ consist of all rules of the form

$$B \to v_0 D_1 v_1 \dots v_{m-1} D_m v_m : N[M[x_1, \dots, x_m]]$$

such that $\mathcal{P}_{\text{right}}$ contains the rule

$$B \to C : N[x_1]$$

and $\mathcal{P}_{\text{left}}$ contains the rule

$$C \to v_0 D_1 v_1 \dots v_{m-1} D_m v_m : M[x_1, \dots, x_m].$$

Let

$$\mathcal{P}' = (\mathcal{P} - \mathcal{P}_{\text{right}}) \cup (\mathcal{P}_{\text{right}} \circ \mathcal{P}_{\text{left}}),$$
$$G' = (\mathcal{N}, \mathcal{T}, \Sigma, \mathcal{P}', S).$$

Lemma 9. $\vdash_{G'} B(w, M)$ *if and only if* $\vdash_G B(w, M)$.

Lemma 10. $B \Rightarrow^+_{G'} B'$ *if and only if* $B \Rightarrow^+_G B'$ *and* $B' \neq C$.

By Lemma 9, $\mathbb{R}(G') = \mathbb{R}(G)$. It is clear that G' is a cycle-free grammar with no ε-rules, and G' has one fewer nonterminals that appear on the right-hand side of unit rules than G. By repeating this procedure, we can obtain a grammar equivalent to G that has no ε- or unit rules.

Left-corner transform Let $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ be a grammar with no ε- or unit rules. Let

$$\mathcal{N}' = \mathcal{N} \cup \{ [B\backslash C] \mid B, C \in \mathcal{N} \},$$

and define $f' : \mathcal{N}' \to \mathrm{Tp}(\mathcal{A})$ by

$$f'(B) = f(B), \qquad f'([B\backslash C]) = f(B) \to f(C).$$

Define $\mathcal{P}'$ as follows:

- For each rule in $\mathcal{P}$ of the form

  $$B \to w_0 B_1 w_1 \dots B_n w_n : M[x_1, \dots, x_n]$$

  ($n \geq 0$) with $w_0 \neq \varepsilon$ and each $C \in \mathcal{N}$, $\mathcal{P}'$ contains the rules

  $$B \to w_0 B_1 w_1 \dots B_n w_n : M[x_1, \dots, x_n],$$
  $$C \to w_0 B_1 w_1 \dots B_n w_n [B\backslash C] : x_{n+1} M[x_1, \dots, x_n].$$

- For each rule in $\mathcal{P}$ of the form

  $$B \to B_1 w_1 \dots B_n w_n : M[x_1, \dots, x_n]$$

  ($n \geq 1$) and each $C \in \mathcal{N}$, $\mathcal{P}'$ contains the rules

  $$[B_1\backslash B] \to w_1 B_2 w_2 \dots B_n w_n : \lambda z.M[z, x_1, \dots, x_{n-1}],$$
  $$[B_1\backslash C] \to w_1 B_2 w_2 \dots B_n w_n \, [B\backslash C] : \lambda z.x_n M[z, x_1, \dots, x_{n-1}]),$$

  (Note that here, either $n \geq 2$ or $w_1 \neq \varepsilon$, since G has no ε- or unit rules.)

Define $G' = (\mathcal{N}', \mathcal{T}, \Sigma, f', \mathcal{P}', S)$. The following lemma implies $\mathbb{R}(G') = \mathbb{R}(G)$.

Lemma 11. *For every $B, D \in \mathcal{N}$ and $w \in \mathcal{T}^+$, the following equivalences hold:*

(i) $\vdash_G B(w, M)$ *if and only if* $\vdash_{G'} B(w, M)$
(ii) $\vdash_G B(Dw, M[x_1])$ *if and only if* $\vdash_{G'} [D\backslash B](w, \lambda z.M[z])$.

Conversion to extended Greibach normal form Let G be a grammar with no ε- or unit rule, and let $G' = (\mathcal{N}', \mathcal{T}, \Sigma, f', \mathcal{P}', S)$ be the result of applying the left-corner transform to G. For each rule π of G', if the left-hand side nonterminal of π is some $B \in \mathcal{N}$, then the right-hand side of π starts with a terminal. If the left-hand side nonterminal of π is of the form $[B\backslash C]$, the right-hand side of π starts either with a terminal or with some nonterminal $B_2 \in \mathcal{N}$. Let $\mathcal{P}'_1$ be the set of all rules in $\mathcal{P}'$ that does not start with a terminal, and for each nonterminal $D \in \mathcal{N}$, let $\mathcal{P}'_D$ be the set of all rules in $\mathcal{P}'$ that has D as their left-hand side nonterminal. If $\pi \in \mathcal{P}'_1$ is of the form $\pi = [B\backslash C] \to D_\pi w_1 B_2 w_2 \dots B_n w_n : M[x_1, \dots, x_n]$, let $\pi \circ \mathcal{P}'_{D_\pi}$ consist of all rules

$$[B\backslash C] \to v_0 E_1 v_1 \dots E_m v_m w_1 B_2 w_2 \dots B_n w_n : M[P[x_1, \dots, x_m], x_{m+1}, \dots, x_{m+n-1}]$$

such that $D_\pi \to v_0 E_1 v_1 \dots E_m v_m : P[x_1, \dots, x_m]$ is a rule in $\mathcal{P}'_{D_\pi}$. Let

$$\mathcal{P}'' = (\mathcal{P}' - \mathcal{P}'_1) \cup \bigcup_{\pi \in \mathcal{P}'_1} \pi \circ \mathcal{P}'_{D_\pi},$$
$$G'' = (\mathcal{N}', \mathcal{T}, \Sigma, f', \mathcal{P}'', S).$$

It is easy to see that $\mathbb{R}(G'') = \mathbb{R}(G')$ and G'' is in *extended Greibach normal form* in the sense that the right-hand side of each rule starts with a terminal.

From extended Greibach normal form to Greibach normal form Let $G = (\mathcal{N}, \mathcal{T}, \Sigma, f, \mathcal{P}, S)$ be a grammar in extended Greibach normal form. Let

$$\mathcal{N}' = \mathcal{N} \cup \{\, [Ba] \mid B \in \mathcal{N}, a \in \mathcal{T} \,\},$$

and define $f' \colon \mathcal{N}' \to \mathrm{Tp}(\mathcal{A})$ by

$$f'(B) = f(B), \qquad f'([Ba]) = f(B) \to f(B).$$

If π is a rule of the form

$$C \to aX_1 \dots X_n : M[x_1, \dots, x_m]$$

in $\mathcal{P}$, where $X_i \in \mathcal{N} \cup \mathcal{T}$, and $k_1, \dots, k_m$ and $j_1, \dots, j_{n-m}$ list the elements of $\{\, i \mid X_i \in \mathcal{N} \,\}$ and $\{\, i \mid X_i \in \mathcal{T} \,\}$, respectively, in increasing order, then let π' be the rule

$$C \to aX'_1 \dots X'_n : x_{j_1}(\dots(x_{j_{n-m}} M[x_{k_1}, \dots, x_{k_m}])\dots),$$

where

$$X'_i = \begin{cases} X_i & \text{if } X_i \in \mathcal{N}, \\ [CX_i] & \text{if } X_i \in \mathcal{T}. \end{cases}$$

Let

$$\mathcal{P}' = \{\, [Ba] \to a : \lambda z.z \mid B \in \mathcal{N}, a \in \mathcal{T} \,\} \cup \{\, \pi' \mid \pi \in \mathcal{P} \,\}.$$

Let $G' = (\mathcal{N}', \mathcal{T}, \Sigma, f', \mathcal{P}', S)$. It is clear that $\mathbb{R}(G') = \mathbb{R}(G)$ and G' is in Greibach normal form.

The constructions in this and the previous subsection together give the second half of the main result of this paper:

Theorem 12. *Given any cycle-free context-free grammar with Montague semantics G such that $\varepsilon \notin \mathbb{L}(G)$, one can construct a Lambek grammar G_{L} such that $\mathbb{R}(G) = \mathbb{R}(G_{\mathrm{L}})$.*

References

1. Kanazawa, M.: Computing interpolants in implicational logics. Annals of Pure and Applied Logic 142, 125–201 (2006)
2. Kanazawa, M., Yoshinaka, R.: Lexicalization of second-order ACGs. NII Technical Report NII-2005-012E, National Institute of Informatics, Tokyo (2005)
3. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly 65, 154–170 (1958)
4. Pentus, M.: Lambek grammars are context free. In: Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science. pp. 429–433 (1993)
5. Pentus, M.: Product-free Lambek calculus and context-free grammars. Journal of Symbolic Logic 62, 648–660 (1997)
6. Pentus, M.: Lambek calculus and formal grammars. In: Provability, Complexity, Grammars, pp. 57–86. No. 192 in American Mathematical Society Translations–Series 2, American Mathematical Society, Providence, Rhode Island (1999)
7. Rosenkrantz, D.J., Lewis II, P.M.: Deterministic left corner parsing. In: IEEE Conference Record of the 11th Annual Symposium on Switching and Automata. pp. 139–152. IEEE (1970)