S4 Enriched Multimodal Categorial Grammars are Context-free

Andrew R. Plummer

Department of Mathematics and Statistics
University Plaza
Georgia State University
Atlanta, Georgia 30303, USA

Abstract

Bar-Hillel et al. [1] prove that applicative categorial grammars weakly recognize the context-free languages. Buszkowski [2] proves that grammars based on the product-free fragment of the non-associative Lambek calculus **NL** recognize exactly the context-free languages. Kandulski [7] furthers this result by proving that grammars based on **NL** also recognize exactly the context-free languages. Jäger [6] proves that categorial grammars based on $\mathbf{NL}\Diamond$, the non-associative Lambek calculus enriched with residuated modalities, weakly recognize exactly the context-free languages. We extend this result, proving that categorial grammars based on $\mathbf{NL}_{\mathrm{S4}}$, the enrichment of $\mathbf{NL}\Diamond$ by the axioms 4 and T, weakly recognize exactly the context-free languages.

1 Introduction

In this paper we consider the generative capacity of a certain class of type-logical grammars. Lambek [9] develops an axiomatic calculus of *syntactic types* that serves as a deductive system upon which grammars recognizing fragments of natural language are predicated. A relation $\rightarrow$ on the set of types is defined as $A \rightarrow B$ if and only if the type A is also of type B. Reflexivity of $\rightarrow$ is immediate. The set of syntactic types $\mathcal{F}$ is built up recursively from a set of atomic types $\mathcal{A}$, the directed implication symbols / and \, and a binary product symbol $\bullet$ as follows:

$$\mathcal{F} ::= \mathcal{A} \mid \mathcal{F}\backslash\mathcal{F} \mid \mathcal{F} \bullet \mathcal{F} \mid \mathcal{F}/\mathcal{F}.$$

The behavior of the logical connectives is governed by the following laws:

$$\begin{array}{rcl} \textit{residuation laws} & : & A \rightarrow C/B \text{ if and only if } A \bullet B \rightarrow C \text{ if and only if } B \rightarrow A\backslash C \\ \textit{law of associativity} & : & (A \bullet B) \bullet C \text{ if and only if } A \bullet (B \bullet C). \end{array}$$

To establish the transitivity of $\rightarrow$,

$$\textit{Cut}\text{: if } A \rightarrow B \text{ and } B \rightarrow C, \text{ then } A \rightarrow C$$

is included in the axiomatic presentation . The calculus above is referred to as the associative Lambek calculus **L**.

Lambek [8], develops a non-associative variant of **L**, referred to as the non-associative Lambek calculus **NL**. That is, **NL** is the calculus **L**, without the *law of associativity*. Moortgat [10] extends the calculi **L** and **NL** to the calculi **L**$\Diamond$ and **NL**$\Diamond$ respectively, by adding to the inventory of logical connectives two unary operators, the unary product $\Diamond$ and the unary slash $\Box^{\downarrow}$. The inferential behavior of the unary operators is governed by the residuation law:

$$\Diamond A \to B \text{ if and only if } A \to \Box^{\downarrow} B.$$

Thus far we have described the pure logic of residuation for the unary operators $\Diamond$ and $\Box^{\downarrow}$, though herein we discuss type-logics enriched with structural rules. Specifically, we consider the following structural postulates:

$K1$:	$\Diamond(A \bullet B) \to \Diamond A \bullet B$	4:	$\Diamond\Diamond A \to \Diamond A$
$K2$:	$\Diamond(A \bullet B) \to A \bullet \Diamond B$	T:	$A \to \Diamond A$.

Before proceeding, we recount some basic definitions. An *alphabet* is a finite set of symbols, denoted by Σ. A *language* is a set of strings over some alphabet. The set of all strings over Σ is denoted by Σ^*. We denote by Σ^+ the set of all strings over Σ, save the null string ϵ.

Definition 1.1. A *context-free grammar* is a quadruple $G = (V, \Sigma, S, P)$, where

V is a finite set of *nonterminal symbols*,
Σ is a finite set, disjoint from V, of *terminal symbols*,
P is a finite set of *productions* of the form $A \to_G \alpha$ ($A \in V$, $\alpha \in (V \cup \Sigma)^*$),
S is an element of V, called the *start symbol*.

Let G be a context-free grammar and let $\to_G^*$ denote the transitive closure of $\to_G$. A string $x \in \Sigma^*$ is *generated* by G if and only if $S \to_G^* x$. The language generated by G is $L(G) = \{x \in \Sigma^* \mid x \text{ is generated by } G\}$. A language L is a *context-free language* if there is a context-free grammar G such that $L = L(G)$. An *ϵ-free context-free grammar* is a context-free grammar with no production of the form $A \to_G \epsilon$. A language L is an *ϵ-free context-free language* if there is an ϵ-free context-free grammar G such that $L = L(G)$. It should be noted that the type-logical grammars discussed herein recognize ϵ-free context-free languages only.

We are interested in situating type-logical grammars within the Chomsky hierarchy. Bar-Hillel et al. [1] prove that applicative categorial grammars weakly recognize the context-free languages. Buszkowski [2] proves that grammars based on the product free fragment of **NL** recognize exactly the context-free languages. Kandulski [7] furthers this result by proving that grammars based on **NL** also recognize exactly the context-free languages. Pentus [11] demonstrates that grammars based on **L** weakly recognize exactly the context-free languages. In Jäger [5] and Jäger [6], it is shown that grammars based on the enriched calculi **L**$\Diamond$ and **NL**$\Diamond$ respectively, also recognize exactly the context-free languages. That is, the enrichment of **L** and **NL** based solely on the residuated unary operators does not increase generative capacity.

Thus it would seem that the generative capacity of grammars based on type-logics is bounded by context-freeness. Yet, Carpenter [3] proves that every recursively enumerable language is recognized by some structurally enriched multimodal categorial grammar. Hence, we are interested in the class of structural rules that increase generative capacity. It is a corollary of results in Emms [4] that $\mathbf{L}\Diamond$ enriched with the interaction postulates $K1$ and $K2$ provides the basis for grammars that recognize non-context-free languages. However, the rules 4 and T remain unanalyzed. In this paper we analyze $\mathbf{NL}\Diamond$ enriched with 4 and T. We establish that this enrichment does not increase the generative capacity of $\mathbf{NL}\Diamond$.

1.1 The Sequent Presentations for NL and NL$\Diamond$

Lambek [9] describes a substructural logic sequent calculus over types. This sequent calculus is equivalent to the axiomatic deductive system **L** in that every sequent derivable via the sequent calculus is derivable from the axioms of **L**. Lambek [9] further proves that **L** has Cut-elimination and the subformula property, and that **L** is decidable. Similar results are obtained for **NL** in Lambek [8], and for $\mathbf{L}\Diamond$ and $\mathbf{NL}\Diamond$ in Moortgat [10].

We present the axiomatic type calculi for **NL** and $\mathbf{NL}\Diamond$, along with their respective sequent calculi. The logical vocabulary of **NL** consists of one binary product $\bullet$ together with its left and right residuation, the directed implications $\backslash$ and $/$. The types of **NL** are defined recursively over some finite alphabet of atomic types $\mathcal{A}$ as

$$\mathcal{F} ::= \mathcal{A}|\mathcal{F}\backslash\mathcal{F}|\mathcal{F}\bullet\mathcal{F}|\mathcal{F}/\mathcal{F}.$$

The behavior of the logical connectives is governed by the following *residuation laws*:

$$A \to C/B \text{ if and only if } A\bullet B \to C \text{ if and only if } B \to A\backslash C.$$

Since **NL** lacks associativity, antecedents of sequents become binary trees via the structural operator $(\cdot,\cdot)$. The set of **NL**-trees is thus given by

$$\mathcal{T} ::= \mathcal{F}|(\mathcal{T},\mathcal{T}).$$

Uppercase Latin letters $A, B, C, \ldots$ are metavariables over types and uppercase Greek letters $\Delta, \Gamma, \Delta', \Gamma', \ldots$ are metavariables over trees of types. The notational convention $\Gamma[\Delta]$ denotes a tree Γ with subtree Δ. When $\Gamma[\Delta]$ is followed in discourse by $\Gamma[\Upsilon]$, we mean that $\Gamma[\Upsilon]$ is the tree $\Gamma[\Delta]$ with the subtree Δ replaced by the tree Υ.

The following Gentzen style sequent presentation provided by Lambek [8] is equivalent to **NL**:

$$\frac{}{A \Rightarrow A}\, id \qquad\qquad \frac{\Delta \Rightarrow A \qquad \Gamma[A] \Rightarrow B}{\Gamma[\Delta] \Rightarrow B}\, Cut$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma[B] \Rightarrow C}{\Gamma[\Delta, A\backslash B] \Rightarrow C}\, \backslash L \qquad\qquad \frac{A,\Gamma \Rightarrow B}{\Gamma \Rightarrow A\backslash B}\, \backslash R$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma[B] \Rightarrow C}{\Gamma[B/A, \Delta] \Rightarrow C}\, /L \qquad\qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow B/A}\, /R$$

$$\frac{\Gamma[A, B] \Rightarrow C}{\Gamma[A \bullet B] \Rightarrow C}\, \bullet L \qquad\qquad \frac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \bullet B}\, \bullet R.$$

Moortgat [10] considers calculi that comprise more than one family of residuated operators, and generalizes the binary operators to the n-ary case. One of the simplest versions of such a multimodal system is the combination of one binary product and its accompanying implications with one unary product and its residuated counterpart. This system is referred to as **NL**$\Diamond$.

The logical vocabulary of **NL**$\Diamond$ is the logical vocabulary of **NL** enriched with two unary connectives, $\Diamond$ and $\Box^{\downarrow}$. The set of **NL**$\Diamond$-types is given by

$$\mathcal{F} ::= \mathcal{A} | \mathcal{F} \backslash \mathcal{F} | \mathcal{F} \bullet \mathcal{F} | \mathcal{F} / \mathcal{F} | \Diamond \mathcal{F} |\, \Box^{\downarrow}\, \mathcal{F}.$$

The unary modalities form a pair of residuated operators. Their logical behavior is governed by the *residuation law*:

$$\Diamond A \to B \text{ if and only if } A \to \Box^{\downarrow} B.$$

We now introduce a unary structural operator $\langle \cdot \rangle$ on trees, occurring in sequent antecedents, corresponding to the unary product $\Diamond$. Therefore, the set of **NL**$\Diamond$-trees is given by

$$\mathcal{T} ::= \mathcal{F} | (\mathcal{T}, \mathcal{T}) | \langle \mathcal{T} \rangle.$$

The following are sequent rules for the unary modalities:

$$\frac{\Gamma[\langle A \rangle] \Rightarrow B}{\Gamma[\Diamond A] \Rightarrow B}\, \Diamond L \qquad\qquad \frac{\Gamma \Rightarrow A}{\langle \Gamma \rangle \Rightarrow \Diamond A}\, \Diamond R$$

$$\frac{\Gamma[A] \Rightarrow B}{\Gamma[\langle \Box^{\downarrow} A \rangle] \Rightarrow B}\, \Box^{\downarrow} L \qquad\qquad \frac{\langle \Gamma \rangle \Rightarrow A}{\Gamma \Rightarrow \Box^{\downarrow} A}\, \Box^{\downarrow} R.$$

The sequent rules of the Gentzen style presentation of **NL**$\Diamond$ are simply the rules of **NL** together with the rules for the unary modalities given above. We write **NL**$\Diamond \vdash \Gamma \Rightarrow A$ if and only if the sequent $\Gamma \Rightarrow A$ is derivable in the **NL**$\Diamond$ sequent calculus. A sequent $\Gamma \Rightarrow A$ such that Γ is an **NL**$\Diamond$-tree and A is a type of **NL**$\Diamond$ is called an **NL**$\Diamond$-*sequent*. The definitions for **NL**-sequents are analogous. Hence, every **NL**-sequent is an **NL**$\Diamond$-sequent.

1.2 NL$\Diamond$-grammars

Definition 1.2. An **NL**$\Diamond$-*grammar* over an alphabet Σ is a pair $\langle \mathcal{L}, \mathcal{D} \rangle$, where $\mathcal{L}$ is a finite relation between Σ^{+} and the set of **NL**$\Diamond$-types $\mathcal{F}$ called a *lexicon*, and $\mathcal{D} \subseteq \mathcal{F}$ is a finite set of *designated* types.

Let $G = \langle \mathcal{L}, \mathcal{D} \rangle$ be an $\mathbf{NL}\diamond$-grammar over an alphabet Σ, $\ell \in \Sigma^+$ and $A \in \mathcal{F}$. If $\langle \ell, A \rangle \in \mathcal{L}$, then ℓ is a *lexical item* corresponding to a *lexical type* A. A string $x \in \Sigma^+$ is recognized by an $\mathbf{NL}\diamond$-grammar if and only if x is a concatenation of lexical items, and replacing each lexical item by one of its corresponding lexical types forms the yield of some binary tree that is the antecedent of a sequent, derivable in $\mathbf{NL}\diamond$, having a designated type as its succedent. This concept is formalized below.

Definition 1.3. Let $G = \langle \mathcal{L}, \mathcal{D} \rangle$ be an $\mathbf{NL}\diamond$-grammar over an alphabet Σ. A string $x = \ell_1 \cdots \ell_n \in \Sigma^+$ is *recognized* by G if and only if there are types $A_1, \ldots, A_n, S$ such that, for all $1 \leq i \leq n$, $\langle \ell_i, A_i \rangle \in \mathcal{L}$, $S \in \mathcal{D}$ and there is a tree Γ with $A_1, \ldots, A_n$ as its yield such that $\mathbf{NL}\diamond \vdash \Gamma \Rightarrow S$.

Let G be an $\mathbf{NL}\diamond$-grammar over an alphabet Σ. The language recognized by G is $\mathfrak{L}(G) = \{x \in \Sigma^+ \mid x \text{ is recognized by } G\}$. A language L is recognized by G if $L = \mathfrak{L}(G)$. The definitions for $\mathbf{NL}$-grammars, $\mathbf{L}$-grammars and $\mathbf{L}\diamond$-grammars are analogous. Recall that Jäger [6] proves the equivalence of the class of context-free languages and the class of languages recognized by $\mathbf{NL}\diamond$-grammars. As noted in the Introduction, the structural interaction rules $K1$ and $K2$ increase generative capacity. Thus, the generative power of a grammar augmented with unary operators is licensed by the axioms of its underlying type calculus. In the remainder of this paper we show that grammars based on $\mathbf{NL}\diamond$ augmented with the structural rules 4 and T still recognize exactly the context-free languages.

2 $\mathbf{NL}_{\mathrm{S4}}$-grammars

2.1 The Type Calculus

We enrich $\mathbf{NL}\diamond$ by adding the following axioms:

$$4 : \diamond\diamond A \to \diamond A \qquad T : A \to \diamond A.$$

We refer to the type calculus $\mathbf{NL}\diamond$ enriched by 4 and T as $\mathbf{NL}_{\mathrm{S4}}$. Notice that $\mathbf{NL}_{\mathrm{S4}}$-types and $\mathbf{NL}_{\mathrm{S4}}$-trees are simply $\mathbf{NL}\diamond$-types and $\mathbf{NL}\diamond$-trees, respectively. The following are sequent rules for 4 and T:

$$\frac{\Gamma[\langle \Delta \rangle] \Rightarrow A}{\Gamma[\langle\langle \Delta \rangle\rangle] \Rightarrow A}\, 4 \qquad\qquad \frac{\Gamma[\langle \Delta \rangle] \Rightarrow A}{\Gamma[\Delta] \Rightarrow A}\, T.$$

The sequent rules of the Gentzen style presentation of $\mathbf{NL}_{\mathrm{S4}}$ are simply the rules of $\mathbf{NL}\diamond$ together with the rules for 4 and T given above. Moortgat [10] proves that the sequent presentation noted above is equivalent to $\mathbf{NL}_{\mathrm{S4}}$. The rules 4 and T are the *structural rules* of $\mathbf{NL}_{\mathrm{S4}}$. The definitions for $\mathbf{NL}_{\mathrm{S4}}$-sequents are analogous to those for $\mathbf{NL}\diamond$. Hence, every $\mathbf{NL}\diamond$-sequent is an $\mathbf{NL}_{\mathrm{S4}}$-sequent. Moortgat [10] proves Cut-elimination, the subformula property, and decidability for $\mathbf{NL}_{\mathrm{S4}}$. The definitions for $\mathbf{NL}_{\mathrm{S4}}$-grammars are analogous to those for $\mathbf{NL}\diamond$.

2.2 Generative Capacity

We now show that $\mathbf{NL}_{S4}$-grammars weakly recognize exactly the context-free languages. We first prove that every context-free language is recognized by some $\mathbf{NL}_{S4}$-grammar. The inclusion of the context-free languages in the class of languages recognized by $\mathbf{NL}_{S4}$-grammars is easily demonstrated. The proof follows almost immediately from the analogous result given in Kandulski [7]. We require the following proposition.

Proposition 2.1. *Let $\Gamma \Rightarrow A$ be an $\mathbf{NL}_{S4}$-sequent containing no modal operators. Then $\mathbf{NL}_{S4} \vdash \Gamma \Rightarrow A$ if and only if $\mathbf{NL} \vdash \Gamma \Rightarrow A$.*

Proof. The sufficiency is obvious, therefore we prove the necessity. Suppose $\mathbf{NL}_{S4} \vdash \Gamma \Rightarrow A$. By the subformula property, $\Gamma \Rightarrow A$ has a proof in which no modal operator occurs. Thus, no sequent appearing in the proof contains $\langle\cdot\rangle$, since each rule introducing $\langle\cdot\rangle$ into a $\langle\cdot\rangle$-free sequent also introduces a modal operator. Hence, $\mathbf{NL} \vdash \Gamma \Rightarrow A$. □

Lemma 2.2. *Every context-free language is recognized by some $\mathbf{NL}_{S4}$-grammar.*

Proof. Let L be a context-free language. Kandulski [7] shows that the class of $\mathbf{NL}$-grammars recognizes exactly the context-free languages. Hence there is an $\mathbf{NL}$-grammar $G = \langle \mathcal{L}, \mathcal{D} \rangle$ that recognizes L. Since neither the lexical nor the designated types contain modal operators, by Proposition 2.1, G recognizes L if G is conceived as an $\mathbf{NL}_{S4}$-grammar. □

Now, to prove that a class of grammars based on a certain type-logic recognize exactly the context-free languages, it is enough to show that a relevant fragment of the type-logic can be axiomatized by finitely many axioms and *Cut*. That is, the fragment is the closure of a finite set of sequents under *Cut*. Pentus [11] utilizes this technique in proving that $\mathbf{L}$-grammars recognize exactly the context-free languages. Jäger [5] and Jäger [6] also utilize this technique in proving that $\mathbf{L}\Diamond$-grammars and $\mathbf{NL}\Diamond$-grammars respectively, recognize exactly the context-free languages. We now provide a brief sketch of the proof technique.

A grammar based on $\mathbf{NL}\Diamond$ contains finitely many types. This implies that the number of connectives appearing in any given type is bounded by some natural number n. The relevant fragment of $\mathbf{NL}\Diamond$ considered for finite axiomatization is the fragment utilizing types containing no more than n connectives. This fragment contains all the types corresponding to strings recognized by the grammar. It is then established that this fragment is axiomatized by sequents having at most two antecedent types, and is closed under *Cut*. Since the fragment contains only finitely many types, it follows immediately that the axiomatization described is finite.

We employ a similar proof technique herein. We show that every $\mathbf{NL}_{S4}$-sequent is derivable in a finitely axiomatizable fragment of $\mathbf{NL}_{S4}$ that is subject to the constraints detailed above. To achieve this, *Cut* must be applicable to any subtree of an $\mathbf{NL}_{S4}$-sequent antecedent. The next lemma (a variation of the interpolation theorem for $\mathbf{L}$) licenses this necessity, and facilitates the desired axiomatization. We also make use of the following definition.

Definition 2.3. Let A and B be types. We define n_c, the number of connectives in a type, as follows:

1. $n_c(A) = 0$ (if A is an atom)
2. $n_c(A \bullet B) = n_c(A/B) = n_c(A\backslash B) = n_c(A) + n_c(B) + 1$
3. $n_c(\Diamond A) = n_c(\Box^{\downarrow} A) = n_c(A) + 1$.

Moreover, let

$$\rho(S) = \begin{cases} n_c(S) & \text{if } S \text{ is a type} \\ max\{n_c(A) \mid A \text{ is a type in } S\} & \text{if } S \text{ is an } \mathbf{NL}_{\mathrm{S4}}\text{-tree or sequent.} \end{cases}$$

Lemma 2.4. *Let $\Gamma[\Delta] \Rightarrow A$ be derivable in $\mathbf{NL}_{\mathrm{S4}}$. Then there is a type B such that $\mathbf{NL}_{\mathrm{S4}} \vdash \Delta \Rightarrow B$, $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma[B] \Rightarrow A$ and $\rho(B) \leq \rho(\Gamma[\Delta] \Rightarrow A)$.*

Proof. Before proceeding, we establish some terminology. If $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma[\Delta] \Rightarrow A$ and a type B possesses the properties given in the statement of the lemma, we call B an *interpolant for* Δ in $\Gamma[\Delta] \Rightarrow A$ (or simply an *interpolant for* Δ if the context is clear). The subtree introduced by a sequent rule into the antecedent of a sequent is called the *active formula*.

We proceed by induction over cut-free sequent derivations. For the base case we consider *id*, in which we simply have $\Gamma = A = B$, and the result is trivial. Therefore, suppose that the result holds for the premises of a sequent rule by which $\Gamma[\Delta] \Rightarrow A$ is inferred. Since sequent derivations are cut-free, it suffices to prove that the result holds for each sequent rule.

The rules $\backslash L$, $\backslash R$, $/L$, $/R$, $\bullet L$, $\bullet R$, $\Diamond L$, $\Diamond R$, $\Box^{\downarrow} L$ and $\Box^{\downarrow} R$ are settled in Jäger [6]. We recount the primary arguments for $\bullet L$ and $\Diamond L$, and fully treat T and 4. We consider three cases concerning the location of the active formula with respect to Δ. That is, either Δ contains the active formula, Δ occurs in the premise of the sequent rule, or Δ does not contain the active formula and does not occur in the premise of the sequent rule.

$$\frac{\Gamma[\Delta'[C, D]] \Rightarrow A}{\Gamma[\Delta'[C \bullet D]] \Rightarrow A} \bullet L \qquad \frac{\Gamma[\Delta'[\langle C \rangle]] \Rightarrow A}{\Gamma[\Delta'[\Diamond C]] \Rightarrow A} \Diamond L$$

Suppose Δ contains the active formula. We first present the argument for $\bullet L$, in which $\Delta = \Delta'[C \bullet D]$. By the induction hypothesis, there is a type B that is an interpolant for $\Delta'[C, D]$. By applying $\bullet L$ to $\Delta'[C, D] \Rightarrow B$, we have $\mathbf{NL}_{\mathrm{S4}} \vdash \Delta'[C \bullet D] \Rightarrow B$. Since $\rho(C), \rho(D) \leq \rho(C \bullet D)$, we have that $\rho(B) \leq \rho(\Gamma[\Delta'[C, D]] \Rightarrow A) \leq \rho(\Gamma[\Delta'[C \bullet D]] \Rightarrow A)$. Hence, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$.

We now present the argument for $\Diamond L$, in which $\Delta = \Delta'[\Diamond C]$. By the induction hypothesis, there is a type B that is an interpolant for $\Delta'[\langle C \rangle]$. By applying $\Diamond L$ to $\Delta'[\langle C \rangle] \Rightarrow B$, we have $\mathbf{NL}_{\mathrm{S4}} \vdash \Delta'[\Diamond C] \Rightarrow B$. Since $\rho(B) \leq \rho(\Gamma[\Delta'[\langle C \rangle]] \Rightarrow A) \leq \rho(\Gamma[\Delta'[\Diamond C]] \Rightarrow A)$, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$.

$$\frac{\Gamma[\Delta'[\langle \Upsilon \rangle]] \Rightarrow A}{\Gamma[\Delta'[\langle \langle \Upsilon \rangle \rangle]] \Rightarrow A} 4 \qquad \frac{\Gamma[\Delta'[\langle \Upsilon \rangle]] \Rightarrow A}{\Gamma[\Delta'[\Upsilon]] \Rightarrow A} T$$

Suppose the rule is T. Thus, $\Delta = \Delta'[\Upsilon]$, and by the induction hypothesis, there is a type B that is an interpolant for $\Delta'[\langle \Upsilon \rangle]$. By applying T to $\Delta'[\langle \Upsilon \rangle] \Rightarrow B$, we have

$\mathbf{NL}_{\mathrm{S4}} \vdash \Delta'[\Upsilon] \Rightarrow B$. Since $\rho(B) \leq \rho(\Gamma[\Delta'[\langle\Upsilon\rangle]] \Rightarrow A) \leq \rho(\Gamma[\Delta'[\Upsilon]] \Rightarrow A)$, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$. Now, suppose the rule is 4. Hence, $\Delta = \Delta'[\langle\langle\Upsilon\rangle\rangle]$, and by similar reasoning, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$.

$$\frac{\Gamma[\langle\Delta'\rangle] \Rightarrow A}{\Gamma[\langle\langle\Delta'\rangle\rangle] \Rightarrow A}\, 4 \qquad \frac{\Gamma[\langle\Delta'\rangle] \Rightarrow A}{\Gamma[\Delta'] \Rightarrow A}\, T$$

Suppose Δ occurs in the premise of the sequent rule. Suppose the sequent rule is T. Then an interpolant for Δ in the premise serves as an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$. We expound the only non-trivial case, $\Delta = \Delta'$. By the induction hypothesis we have a type B that is an interpolant for Δ'. By applying T to $\Gamma[\langle B\rangle] \Rightarrow A$, we have $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma[B] \Rightarrow A$. Hence, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$. If the sequent rule is 4, then again, an interpolant for Δ in the premise serves as an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$.

Suppose Δ does not contain the active formula and does not occur in the premise of the sequent rule. Then the sequent rule must be 4, and $\Delta = \langle\langle\Delta'\rangle\rangle$. By the induction hypothesis, there is a type B that is an interpolant for $\langle\Delta'\rangle$. By applying 4 to $\langle\Delta'\rangle \Rightarrow B$ we have $\mathbf{NL}_{\mathrm{S4}} \vdash \langle\langle\Delta'\rangle\rangle \Rightarrow B$. Since, $\rho(\Gamma[\langle\Delta'\rangle] \Rightarrow A) = \rho(\Gamma[\langle\langle\Delta'\rangle\rangle] \Rightarrow A)$, B is an interpolant for Δ in $\Gamma[\Delta] \Rightarrow A$. □

We define a *deductive sequent system* to be a set of sequents $\Gamma \Rightarrow A$ which is closed under Cut. A deductive sequent system is finitely axiomatizable if and only if it is the closure of a finite set of sequents under Cut. The following definition describes our desired finite axiomatization. We then prove that the relevant fragment of $\mathbf{NL}_{\mathrm{S4}}$ is derivable within the deductive sequent system.

Definition 2.5. For any non-negative integer n, the deductive sequent system P_n is the closure of the following set of axioms under Cut: $\{A \Rightarrow B \mid \mathbf{NL}_{\mathrm{S4}} \vdash A \Rightarrow B$ and $\rho(A), \rho(B) \leq n\} \cup \{\langle A\rangle \Rightarrow B \mid \mathbf{NL}_{\mathrm{S4}} \vdash \langle A\rangle \Rightarrow B$ and $\rho(A), \rho(B) \leq n\} \cup \{(A, B) \Rightarrow C \mid \mathbf{NL}_{\mathrm{S4}} \vdash (A, B) \Rightarrow C$ and $\rho(A), \rho(B), \rho(C) \leq n\}$.

Lemma 2.6. *Let $\Gamma \Rightarrow A$ be an $\mathbf{NL}_{\mathrm{S4}}$-sequent. If $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma \Rightarrow A$ and $\rho(\Gamma \Rightarrow A) \leq n$, then $\Gamma \Rightarrow A \in P_n$.*

Proof. We proceed by induction over the number of structural operators, $(\cdot, \cdot)$ and $\langle\cdot\rangle$, in Γ. If Γ contains no structural operators then Γ is a single type and the result is trivial. Therefore, assume that Γ contains at least one structural operator. We consider two cases. Suppose $\Gamma = \Delta[\langle C\rangle]$, where C is a type. By Lemma 2.4, there is an interpolant B for $\langle C\rangle$ in $\Delta[\langle C\rangle] \Rightarrow A$. Since $\rho(C) \leq n$, by Definition 2.5, $\langle C\rangle \Rightarrow B \in P_n$. Since $\rho(\Delta[B] \Rightarrow A) \leq n$, by the induction hypothesis $\Delta[B] \Rightarrow A \in P_n$. Applying Cut to the premises $\langle C\rangle \Rightarrow B$ and $\Delta[B] \Rightarrow A$, we have that $\Delta[\langle C\rangle] \Rightarrow A \in P_n$. Now, suppose $\Gamma = \Delta[(C, D)]$, where C and D are types. By Lemma 2.4, there is an interpolant B for (C, D) in $\Delta[(C, D)] \Rightarrow A$. Since $\rho(C), \rho(D) \leq n$, by Definition 2.5, $(C, D) \Rightarrow B \in P_n$. Since $\rho(\Delta[B] \Rightarrow A) \leq n$, by the induction hypothesis $\Delta[B] \Rightarrow A \in P_n$. Applying Cut to the premises $(C, D) \Rightarrow B$ and $\Delta[B] \Rightarrow A$, we have that $\Delta[(C, D)] \Rightarrow A \in P_n$. □

Based on the axiomatization P_n, we may now prove the main result.

Lemma 2.7. *Every language recognized by an* $\mathbf{NL}_{\mathrm{S4}}$*-grammar is context-free.*

Proof. Let $G = \langle \mathcal{L}, \mathcal{D} \rangle$ be an $\mathbf{NL}_{\mathrm{S4}}$-grammar over an alphabet Σ and let $n = max\{\rho(A) \mid A$ is a type occurring in $G\}$. We construct an equivalent context-free grammar G' in the following way. The terminal symbols of G' are the lexical items of G. The nonterminal symbols of G' are the $\mathbf{NL}_{\mathrm{S4}}$-types A such that $\rho(A) \leq n$. By relabeling if necessary, S is the start symbol of G'. The productions of G' are $\{S \rightarrow_{G'} D \mid D \in \mathcal{D}\} \cup \{A \rightarrow_{G'} B \mid B \Rightarrow A \in P_n\} \cup \{A \rightarrow_{G'} B \mid \langle B \rangle \Rightarrow A \in P_n\} \cup \{A \rightarrow_{G'} BC \mid (B, C) \Rightarrow A \in P_n\} \cup \{A \rightarrow_{G'} \ell \mid \langle \ell, A \rangle \in \mathcal{L}\}$.

Suppose $\ell_1 \dots \ell_k \in \mathfrak{L}(G)$. Then there are types $A_1, \dots, A_k, D$ such that, for all $1 \leq i \leq k$, $\langle \ell_i, A_i \rangle \in \mathcal{L}$, $D \in \mathcal{D}$ and there is a tree Γ with $A_1, \dots, A_k$ as its yield such that $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma \Rightarrow D$. By the construction of G', $S \rightarrow_{G'} D$ and $A_i \rightarrow_{G'} \ell_i$ for each i. By Lemma 2.6, $D \rightarrow^*_{G'} A_1 \dots A_k$. Since the production relation $\rightarrow_{G'}$ is transitive, it follows that $S \rightarrow^*_{G'} \ell_1 \dots \ell_k$. Hence, $\ell_1 \dots \ell_k \in L(G')$.

Now, suppose $\ell_1 \dots \ell_k \in L(G')$. That is, $S \rightarrow^*_{G'} \ell_1 \dots \ell_k$. By the construction of G', we must have $D \in \mathcal{D}$ and $A_1, \dots, A_k$ with $\langle \ell_i, A_i \rangle \in \mathcal{L}$ such that $D \rightarrow^*_{G'} A_1 \dots A_k$. Hence, there is a derivation tree Γ with D as its root node and $A_1, \dots, A_k$ as its yield. Since the productions of G' correspond to $\mathbf{NL}_{\mathrm{S4}}$-sequents in P_n, and since all sequents in P_n are, by definition, derivable in $\mathbf{NL}_{\mathrm{S4}}$, it follows that $\mathbf{NL}_{\mathrm{S4}} \vdash \Gamma \Rightarrow D$. Thus, $\ell_1 \dots \ell_k \in \mathfrak{L}(G)$. □

Notice that the context-free grammar simulating the $\mathbf{NL}_{\mathrm{S4}}$-grammar completely ignores the $\langle \cdot \rangle$ structures of an $\mathbf{NL}_{\mathrm{S4}}$-tree. This follows from the fact that, due to rule T, in derivations, we may employ only those sequents without $\langle \cdot \rangle$ structure.

Theorem 2.8. $\mathbf{NL}_{\mathrm{S4}}$*-grammars recognize exactly the context-free languages.*

Proof. Immediate from Lemma 2.2 and Lemma 2.7. □

3 Conclusion

This article shows that enriching the type calculus $\mathbf{NL}\Diamond$ with the structural postulates 4 and T does not increase its generative capacity. To achieve this result we utilize a proof employed in Jäger [6]. The adapted proof is based on a variation of the interpolation lemma for $\mathbf{L}$, and a finitely axiomatizable set of $\mathbf{NL}_{\mathrm{S4}}$-sequents comprising no more than two antecedent types. Moreover, it is shown that simple structural rules are capable of extending the generative capacity of $\mathbf{NL}\Diamond$, necessitating the study of structurally enriched variants of type-logical grammars. Specifically, we submit for further research, an extension of the result of Emms [4], establishing that the structural rules $K1$ and $K2$ extend the generative capacity of $\mathbf{NL}\Diamond$. Furthermore, we are interested in the generative capacity of $\mathbf{L}\Diamond$ enriched with the structural postulates 4 and T. Generally, we state that the generative capacity of structurally enriched multimodal grammars is largely unstudied. The proof strategy employed in this paper remains viable with respect to other type-logical grammars further enriched with additional structural postulates. As stated, certain structural rules extend the generative capacity of grammars based on variants of $\mathbf{L}$. It is recommended that further results be established regarding the generative capacity of structurally enriched multimodal categorial grammars.

4 Acknowledgements

The author wishes to thank the referees and the editor for their encouragement, and for their insightful comments regarding the content and presentation of this paper.

References

[1] Y. Bar-Hillel, C. Gaifman, E. Shamir, On categorial and phrase structure grammars, *Bulletin of the Research Council of Israel* **F**(9), (1960) 1–16.

[2] W. Buszkowski, Generative capacity of non-associative Lambek calculus, *Bulletin of the Polish Academy of Sciences: Mathematics* **34**, (1986) 507–518.

[3] B. Carpenter, The Turing-completeness of multi-modal categorial grammars, Papers presented to Johan van Benthem in honor of his 50th birthday, European Summer School in Logic, Language and Information, (1999) Utrecht.

[4] M. Emms, Extraction Covering Extensions of Lambek calculus are not context-free, in *Proceedings of the Ninth Amsterdam Colloquium*, P. Dekker, M. Stokhof, ed., (University of Amsterdam, 1994) 269–286.

[5] G. Jäger, On the generative capacity of multi-modal categorial grammars, *Research on Language and Computation* **1**, (2003) 105–125.

[6] G. Jäger, Residuation, structual rules and context freeness, *Journal of Logic, Language and Information* **13**, (2004) 47–59.

[7] M. Kandulski, The equivalence of non-associative Lambek categorial grammars and context-free grammars, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **34**, (1988) 41–52.

[8] J. Lambek, On the calculus of syntatic types, in *Structure of Language and Its Mathematical Aspects*, R. Jacobson, ed., (Amer. Math. Soc., Providence, RI 1961) 166–178.

[9] J. Lambek, The mathematics of sentence structure *American Mathematical Monthly* **65**, (1958) 154–170.

[10] M. Moortgat, Multimodal Linguistic Inference, *Journal of Logic, Language and Information* **5**, (1996) 349–385.

[11] M. Pentus, Lambek grammars are context free, *Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science*, Montreal (1993).