Residuation, Structural Rules and Context Freeness

Gerhard Jäger
University of Potsdam/ZAS Berlin

1. Introduction

This paper deals with the issue of the generative capacity of a certain version of type logical categorial grammar. Originally categorial grammar in the modern sense was invented in three varieties in the late fifties and early sixties. Bar-Hillel (1953) developed applicative categorial grammar, a bidirectional version of older type theoretic systems tracing back to the work of Polish logicians in the early twentieth century. Few years later, Lambek (1958) proposed his calculus of syntactic types that is known as the (associative) "Lambek calculus" nowadays (abbreviated as "Ł"). A short time later he published a non-associative version of this calculus in (Lambek, 1961), which is known as the "Non-associative Lambek calculus" **NL**. These two systems are the first instances of "Type Logical Grammars", i.e. the deductive machinery of the grammar formalism is a substructural type logical calculus.

The issue of the position of these grammar formalisms within the Chomsky hierarchy has intrigued mathematical linguists from the beginning. It was settled first for applicative categorial grammar by Bar-Hillel, Gaifman and Shamir (1960). They establish the weak equivalence of this version of categorial grammars with the context free languages.

For the type logical categorial grammars, this problem was settled fairly late. Buszkowski (1986) established the the product free fragment of the non-associative Lambek calculus defines exactly the context free languages, and Kandulski (1988) showed that this result carries over to full **NL**. Finally, (Pentus, 1993) gives a proof that the associative Lambek calculus Ł is weakly equivalent to the context free grammars as well.

It was already conjectured in (Chomsky, 1957) that context free grammars are not expressive enough to give an adequate description of the grammar of natural languages. This was formally proved in (Shieber, 1985). So it seems that none of the tree basic varieties of categorial grammar provides an adequate grammar formalism for linguistics. This shortcoming motivated to move to multimodal systems, i.e. type logical grammars that employ several families of connectives and certain interactions between them. This idea was first explored in (Moortgat, 1988) and (Morrill, 1990), and systematized in (Moortgat, 1996). There it is assumed that it is sufficient to use a finite ensemble of residuation connectives plus some interaction postulates between them to come to terms with the empirical facts of natural language. This idea is confirmed but trivialized by (Carpenter, 1999), where it is proved that multimodal type logical grammars are equivalent in generative power to Turing machines.

Considering Carpenter's proof, it seems intuitively obvious that the unrestricted use of interaction postulates is responsible for this increase in generative capacity, while the notion of multimodality as such has no such effect. This is partially confirmed by Jäger (2001). This article gives a proof that enriching the associative Lambek calculus with pairs of unary residuation connectives without interaction postulates does not increase generative power; the resulting system still describes exactly the context free languages. This is established by a straightforward extension of Pentus' construction.

For a non-associative base logic, several important results in this connection have been obtained by Maciej Kandulski (see (Kandulski, 1995; Kandulski, 2002)). He generalizes his result from (Kandulski, 1988) to the commutative version of the non-associative Lambek calculus, and to multimodal logics comprising an arbitrary number of different families of residuation connectives of any arity, but without structural rules.

Kandulski's results are all based on an axiomatization of the type logics underlying the grammars in question and a process of proof normalization within this axiomatic calculus. The present paper presents alternative proofs of these theorems that are based on the Gentzen style sequent presentation of the logics involved. This new proof strategy leads to generalizations of Kandulski's results in two respects: Any combination of residuated connectives with any of the structural rules Permutation, Contraction and Expansion lead to type logical grammars that recognized only context free languages, and this also holds if we admit non-atomic designated types.

The structure of the paper is as follows. We first focus on the simplest multimodal extension of the non-associative Lambek calculus, namely the calculus **NL**$\Diamond$ from (Moortgat, 1996). In section 2 we introduce the necessary technical notions, and section 3 presents the proof that grammars based on **NL**$\Diamond$ recognized exactly the context free languages. In section 4 we generalize these results to residuation modalities of arbitrary arity, and to

 Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+6), pp. 151–158. Universitá di Venezia.

calculi using the structural rules Permutation, Contraction or Expansion. Section 5 summarizes the findings and points to desiderata for further research.

2. Technical Preliminaries

2.1. NL

The non-associative Lambek calculus **NL** is the weakest substructural logic. Its logical vocabulary consists of one binary product $\bullet$ and its left and right residuation, the two directed implications $\backslash$ and $/$. More formally, 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 calculus itself can be defined as a set of arrows, i.e. objects of the form $A \to B$, where A and B are types of **NL**. In its axiomatic presentation, the calculus comprises the identity axiom and the Cut rule. We use the upper case Latin letter $A, B, C, ...$ as meta-variables over types.

$$\frac{}{A \to A}\, id$$

$$\frac{A \to B \qquad B \to C}{A \to C}\, Cut$$

The behavior of the logical connectives is governed by the residuation laws

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

Lambek also gives a Gentzen style sequent presentation of **NL**. A sequent consists of an antecedent and a succedent, where the antecedent is a binary tree over types and the succedent a single type. We write trees as terms formed from types and the binary operation $\circ$. Formally, the set of **NL**-trees is thus given by

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

The upper case Latin letters $X, Y, Z, ...$ are meta-variables over trees of types. $X[Y]$ is a tree X containing a sub-tree Y, and $X[Z]$ is the result of replacing the sub-tree Y in X by Z.

$$\frac{}{A \Rightarrow A}\, id \qquad \frac{X \Rightarrow A \qquad Y[A] \Rightarrow B}{Y[X] \Rightarrow B}\, Cut$$

$$\frac{X[A \circ B] \Rightarrow C}{X[A \bullet B] \Rightarrow C}\, \bullet L \qquad \frac{X \Rightarrow A \qquad Y \Rightarrow B}{X \circ Y \Rightarrow A \bullet B}\, \bullet R$$

$$\frac{X \Rightarrow A \qquad Y[B] \Rightarrow C}{Y[B/A \circ X] \Rightarrow C}\, /L \qquad \frac{X \circ A \Rightarrow B}{X \Rightarrow A/B}\, /R$$

$$\frac{X \Rightarrow A \qquad Y[B] \Rightarrow C}{Y[X \circ A\backslash B] \Rightarrow C}\, \backslash L \qquad \frac{A \circ X \Rightarrow B}{X \Rightarrow A\backslash B}\, \backslash R$$

Figure 1: Sequent presentation of the non-associative Lambek calculus **NL**

We write "**NL** $\vdash X \Rightarrow A$" iff the sequent $X \Rightarrow A$ is derivable in the sequent calculus. The axiomatic and the sequent presentation of **NL** are equivalent in the sense that every derivable arrow is also a derivable sequent, and replacing all occurrences of $\circ$ in a derivable sequent by the product $\bullet$ yields a derivable arrow.

It is easy to see that all rules in the sequent calculus except Cut have the subformula property. Lambek proved Cut elimination for the sequent calculus, which establishes decidability.

2.2. NL$\Diamond$

(Moortgat, 1996) extends the format of type logics in two ways. He considers calculi that comprise more than one family of residuated operators, and he generalizes Lambek's binary operators to the n-ary case. In the present paper, we will be mainly concerned with one of the simplest version of such a multimodal system, namely the combination of one binary product and its accompanying implications with one unary product and its residuated counterpart. The resulting logic is dubbed **NL**$\Diamond$.

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

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

They form a pair of residuated operators, i.e. their logical behavior is governed by the residuation law

$$A \to \Diamond B \text{ iff } \Box^{\downarrow} A \to B$$

The axiomatic presentation of **NL**$\Diamond$consists just of the axioms and rules of **NL** plus the above residuation law.

(Moortgat, 1996) also gives a sequent presentation of **NL**$\Diamond$. Now the trees that occur in the antecedent of a sequent is composed from types by two operators, a binary one ($\circ$), and a unary one ($\langle\cdot\rangle$), corresponding to the two products $\bullet$ and $\Diamond$. So we have

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

Moortgat's sequent calculus for **NL**$\Diamond$ is obtained by extending the sequent calculus for **NL** with the following four rules, i.e. a rule of use and a rule of proof for both $\Diamond$and $\Box^{\downarrow}$.

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

$$\frac{X[A] \Rightarrow B}{X[\langle\Box^{\downarrow} A\rangle] \Rightarrow B}\Box^{\downarrow} R \qquad \frac{\langle X\rangle \Rightarrow A}{X \Rightarrow \Box^{\downarrow} A}\Box^{\downarrow} R$$

Figure 2: Sequent rules for the unary modalities in **NL**$\Diamond$

As in **NL**, all sequent rules of **NL**$\Diamond$ have the subformula property. By proving Cut elimination for **NL**$\Diamond$, (Moortgat, 1996) thus establishes decidability, and he also proves the equivalence of the axiomatic with the sequent presentation.

2.3. Logic and grammars

A type logic like **NL**$\Diamond$ is the deductive backbone of a type logical grammar. The grammar itself consists just of the lexicon, i.e. an assignment of types to lexical items, and a collection of designated types (which is sometimes tacitly assumed to be the singleton set $\{s\}$, but I assume a more general notion of grammar here).

Definition 1 (NL$\Diamond$-grammar) *An **NL**$\Diamond$-grammar over an alphabet Σ is a pair $\langle\mathcal{L}, \mathcal{D}\rangle$, where $\mathcal{L}$, the lexicon, is a finite relation between Σ^+ and the set of **NL**$\Diamond$-types $\mathcal{F}$, and the set of designated types $\mathcal{D}$ is a finite subset of $\mathcal{F}$.*

(The definitions for "**NL**-grammar", "Ł-grammar" etc. are identical.) A string from Σ^+ is recognized by an **NL**$\Diamond$-grammar G if it is a concatenation of lexical items, and replacing each of these items by one of their lexical types leads to the yield of some binary tree of types from which a designated category is derivable. Formally this reads as follows:

Definition 2 (Recognition) *Let $G = \langle\mathcal{L}, \mathcal{D}\rangle$ be an **NL**$\Diamond$-Grammar over Σ. A string $w \in \Sigma^+$ is recognized by G iff*

- $w = v_1 \cdots v_n$,
- *there are $A_1, \ldots, A_n \in \mathcal{F}$ such that for all $1 \leq i \leq n : \langle v_i, A_i\rangle \in \mathcal{L}$,*

- *there is a tree X and a type $S \in \mathcal{D}$ such that **NL**$\Diamond \vdash X \Rightarrow S$, and $A_1 \ldots A_n$ is the yield of X.*

The notion of recognition for other categorial calculi is similar—the only difference being the underlying calculus and thus the derivability relation in the last clause.

3. Generative Capacity

In this section I will present and discuss the main result of this paper, the weak generative equivalence between context free grammars and **NL**$\Diamond$-grammars. The inclusion of the context free languages in the class of **NL**$\Diamond$-recognizable languages is easy to show; the proof immediately follows from Kandulski's (1988) analogous proof for **NL** (which is itself a straightforward adaption of the corresponding proof for applicative categorial grammars from (Bar-Hillel, Gaifman and Shamir, 1960)).

Lemma 1 *Every context free language L is recognized by some **NL**$\Diamond$-grammar.*

Proof: Kandulski (1988) proves that the class of **NL**-grammars recognizes exactly the context free languages. Thus there is an **NL**-Grammar $G = \langle \mathcal{L}, \mathcal{D} \rangle$ that recognizes L. From the facts that all sequent rules of cut-free **NL**$\Diamond$ have the subformula property and that all sequent rules of **NL** are also rules of **NL**$\Diamond$ it follows that **NL**$\Diamond$ is a conservative extension of **NL**. In other words, if an **NL**$\Diamond$-sequent $X \Rightarrow A$ is derivable in **NL**$\Diamond$ and does not contain occurrences of $\Diamond, \Box^{\downarrow}$, and $\langle \cdot \rangle$, it is also **NL**-derivable. The structural connective $\langle \cdot \rangle$ only cooccurs with the modalities $\Diamond$ or $\Box^{\downarrow}$ in derivable sequents. (This can easily be shown by induction over sequent derivations.) So if a derivable **NL**$\Diamond$-sequent does neither contain $\Diamond$ nor $\Box^{\downarrow}$, it is also **NL**-derivable. To decide whether a string is recognized by G or not it is sufficient to restrict attention to sequents that only involve types from $\mathcal{L}$ or $\mathcal{D}$. For this fragment, the derivability relations defined by **NL** and **NL**$\Diamond$ coincide. Therefore G still recognizes L if it is conceived as an **NL**$\Diamond$-grammar. $\dashv$

To prove that a given variety of type logical grammar recognizes only context free languages, it is sufficient to show that the relevant fragment of the underlying logic can be axiomatized by using only finitely many axioms and the Cut rule. (This strategy has in fact been pursued in all such proofs from the literature that were mentioned in the introduction.) Pentus (1993) proof for the context freeness of Ł is an especially simple and elegant implementation of this idea (even though the proof for the correctness of his construction is quite complex). Consider an Ł-grammar $G = \langle \mathcal{L}, \mathcal{D} \rangle$. It comprises finitely many types (either as lexical or as designated types), and hence there is some upper limit n for the complexity of types in G, where the complexity of a type is identified with the number of connectives occurring in it. The first important insight of his proof is that to one does not need the entire calculus Ł but just those fragment of it that only uses types with a complexity $\leq n$ to determine which language G recognizes. Let us call this fragment of Ł Ł(n). The central lemma of the proof establishes that Ł(n) can be axiomatized by the set of its sequents that have at most two antecedent types, and the Cut rule. Since there are only finitely many types occurring in Ł(n), this set of axiom is finite.

The same construction can be applied to **NL**$\Diamond$ as well. The adequacy of the construction is in fact much easier to prove here then for Ł. We first show that every **NL**$\Diamond$-sequent can be represented as the result of Cut in such a way that the premises of this Cut rule do not involve types that are more complex than the most complex type in the original sequent. Furthermore, any position in the original sequent can be chosen as the target position for the Cut application.

Lemma 2 *Let $X[Y] \Rightarrow A$ be a theorem of **NL**$\Diamond$. Then there is a type B such that*

1. ***NL**$\Diamond \vdash Y \Rightarrow B$*
2. ***NL**$\Diamond \vdash X[B] \Rightarrow A$*
3. *There is a type occurring in $X[Y] \Rightarrow A$ which contains at least as many connectives as B.*

Proof: We prove the lemma by induction over sequent derivations. For the base case id the lemma is obviously true. So let us suppose the lemma holds for the premises of a sequent rule. We have to demonstrate that it holds for the conclusion as well. If **NL**$\Diamond \vdash X[Y] \Rightarrow A$ and B has the required properties, we call B a *witness for* Y (with respect to the sequent $X[Y] \Rightarrow A$).

Suppose that $X[Y] \Rightarrow A$ is a premise of a sequent rule and $X'[Y'] \Rightarrow B$ is the conclusion, where Y' is the substructure corresponding to Y. Then for any type C, if $\vdash Y \Rightarrow C$, then $\vdash Y' \Rightarrow C$. (Either Y and Y' are

identical, or Y' is the result of applying a rule of use to Y, which is derivability preserving.) Furthermore, if $\vdash X[D] \Rightarrow A$ for some type D, then $\vdash X'[D] \Rightarrow B$ as well. (Either $X[D]$ and $X'[D]$ are simply identical, or X' is the result of applying a rule of use to X with Y as an inactive part. In the latter case, it does not matter for derivability if we replace Y with D.) Finally (Moortgat, 1996) proves that the sequent calculus of **NL**$\Diamond$ enjoys the subformula property. Thus if B is a witness for Y in the premise, it is also a witness for Y' in the conclusion (because the most complex type from the premise occurs in the conclusion as subtype). So to complete the induction step, we only have to consider cases where a substructure in the conclusion does not correspond to any substructure in any of the premises of a sequent rule. This applies to all types that are created via a left introduction rule. It is obvious though that each type is a witness for itself, so the induction step holds for these cases as well. So the only cases that remain to be considered are the left hand sides of the conclusions in $\bullet R$ and $\Diamond R$. In either case, the type on the right hand side is a witness for the left hand side as a whole. This completes the proof. $\dashv$

From this it follows immediately that a Pentus style axiomatization is possible for **NL**$\Diamond$ as well.

Lemma 3 *Let* ***NL***$\Diamond(n) = \{A \Rightarrow B | \mathbf{NL}\Diamond \vdash A \Rightarrow B \& max(\#A, \#B) \leq n\} \cup \{\langle A\rangle \Rightarrow B | \mathbf{NL}\Diamond \vdash \langle A\rangle \Rightarrow B \& max(\#A, \#B) \leq n\} \cup \{A \circ B \Rightarrow C | \mathbf{NL}\Diamond \vdash A \circ B \Rightarrow C \& max(\#A, \#B, \#C) \leq n\}$*, where $\#A$ is the number of connectives occurring in A. Furthermore, let $X \Rightarrow A$ be an* ***NL***$\Diamond$*-derivable sequent such that no type in it contains more than n connectives. Then $X \Rightarrow A$ is derivable from* ***NL***$\Diamond(n)$ *and Cut.*

Proof: We prove the lemma by induction over the number of structural operators (i.e. $\circ$ and $\langle\cdot\rangle$) in X. If X is a single type, the lemma is obviously true. So let us assume that $X = Y[Z]$, where $Z = B_1 \circ B_2$ or $Z = \langle B\rangle$. According to lemma 2, there is a witness C for Z such that $\vdash Z \Rightarrow C, Y[C] \Rightarrow A$, and $\#C \leq n$. Then $Z \Rightarrow C \in$ **NL**$\Diamond(n)$ by definition and $Z \Rightarrow C$ is derivable from **NL**$\Diamond(n)$ and Cut by induction hypothesis. Thus $X \Rightarrow A$ is derivable from **NL**$\Diamond(n)$ and Cut as well. $\dashv$

This leads directly to the inclusion of the class of **NL**$\Diamond$-recognizable languages in the class of context free languages.

Lemma 4 *Every* ***NL***$\Diamond$*-recognizable language is context free.*

Proof: Let an **NL**$\Diamond$-grammar $G_1 = \langle \mathcal{L}, \mathcal{D}\rangle$ (with $\mathcal{L}$ being the lexicon and $\mathcal{D}$ the set of designated types) be given. We construct an equivalent CFG G_2 in the following way: The terminal elements of G_2 are the lexical items of G_1. The non-terminals are all **NL**$\Diamond$-types A with $\#A \leq n$, where n is the maximal number of connectives in a single type occurring in G_1. Besides we have a fresh non-terminal S which is the start symbol. Productions are

- $\{A \rightarrow B | B \Rightarrow A \in \mathbf{NL}\Diamond(n)\} \cup$
- $\{A \rightarrow B | \langle B\rangle \Rightarrow A \in \mathbf{NL}\Diamond(n)\} \cup$
- $\{A \rightarrow B, C | B \circ C \Rightarrow A \in \mathbf{NL}\Diamond(n)\} \cup$
- $\{A \rightarrow v | \langle v, A\rangle \in \mathcal{L}\} \cup$
- $\{S \rightarrow A | A \in \mathcal{D}\}$

If $v_1 \ldots v_m$ is recognized by G_1, then there is a **NL**$\Diamond$-derivable sequent $X \Rightarrow B$ such that B is a designated category, the yield of X is $A_1 \ldots A_m$, and $\langle v_i, A_i\rangle \in \mathcal{L}$ for $1 \leq i \leq m$. By the construction of G_2, $S \rightarrow^*_{G_2} B$. Due to lemma 3 and the construction of G_2, thus $S \rightarrow^*_{G_2} A_1 \ldots A_n$, and by the construction of G_2, this leads to $S \rightarrow^*_{G_2} v_1 \ldots v_n$. So $v_1 \ldots v_n$ is recognized by G_2.

No suppose $v_1 \ldots v_m$ is recognized by G_2. This means that $S \rightarrow^*_{G_2} v_1 \ldots v_n$. By the construction of G_2, there must be a $B \in \mathcal{D}$ and $A_1 \ldots A_n$ with $\langle v_i, A_i\rangle \in \mathcal{L}$ such that $B \rightarrow^*_{G_2} A_1 \ldots A_n$. Hence there must be a derivation from B to some structure X such that $A_1 \ldots A_n$ is the yield of X. All rules involved in this derivation originate from **NL**$\Diamond(n)$, and since all rules in **NL**$\Diamond(n)$ are **NL**$\Diamond$-derivable, $\vdash X \Rightarrow B$. Hence $v_1 \ldots v_n$ is recognized by G_1. $\dashv$

The Lemmas 1 and 4 jointly give the main result of this section:

Theorem 1 ***NL***$\Diamond$*-grammars recognize exactly the context free languages.*

Proof: Immediate. $\dashv$

4. Generalizations

The concept of residuated logical connectives can readily be generalized to n-ary operations for arbitrary n. Such this systems have been considered at various places in the context of categorial grammar, including (Moortgat, 1996) and (Kandulski, 2002). A multimodal logic of pure residuation ("**LPR**" henceforth) is characterized by a family of modes $\mathcal{M}$ and a function δ that assigns each mode an arity, i.e. a natural number. If $f \in \mathcal{M}$ is a mode of arity $\delta(f) = m$, it defines $m + 1$ m-ary connectives: an m-ary product operator $f_\bullet$, and m implications $\{f^i_\rightarrow | 1 \leq i \leq m\}$. As for **NL** and **NL**$\diamond$, there is an axiomatic formulation for any **LPR** having the identity axiom as only axiom, the Cut rule and a collection of residuation laws. The laws for the binary and unary operators given above are generalized to the general case in the following way:

$$\forall f \forall i \leq \delta(f) : f_\bullet(A_1, \cdots, A_{\delta(f)}) \rightarrow B \text{ iff } A_i \rightarrow f^i_\rightarrow(A_1, \ldots, A_{i-1}, B, A_{i+1}, \ldots, A_{\delta(f)})_i$$

The sequent calculus for a given **LPR** over the set of modes $\mathcal{M}$ is also a straightforward extrapolation from **NL**$\diamond$. Antecedents of sequents are now terms built from types by means of structural operators. There is one structural operator $f_\circ$ for every $f \in M$ with arity $\delta(f)$.

$$\frac{}{A \Rightarrow A}\, id \qquad \frac{X \Rightarrow A \qquad Y[A] \Rightarrow B}{Y[X] \Rightarrow B}\, Cut$$

$$\frac{\forall i \leq \delta(f) : X_i \Rightarrow A_i}{f_\circ(X_1, \cdots, X_{\delta(f)}) \Rightarrow f_\bullet(A_1, \cdots, A_{\delta(f)})}\, f_\bullet L \qquad \frac{X[f_\circ(A_1, \cdots, A_{\delta(f)})] \Rightarrow B}{X[f_\bullet(A_1, \cdots, A_{\delta(f)})] \Rightarrow B}\, f_\bullet R$$

$$\frac{\forall i \leq \delta(f), i \neq j : X_i \Rightarrow A_i \qquad Y[B] \Rightarrow C}{Y[f_\circ(X_1, \cdots, X_{i-1}, f_\rightarrow(A_1, \cdots, A_{i-1}, B, A_{i+1} \cdots, A_{\delta(f)}), X_{i+1}, \cdots, X_{\delta(f)})] \Rightarrow C}\, f_\rightarrow L$$

$$\frac{f_\circ(A_1, \cdots, A_{i-1}, X, A_{i+1}, \cdots, A_{\delta(f)}) \Rightarrow B}{X \Rightarrow f_\rightarrow(A_1, \cdots, A_{i-1}, B, A_{i+1}, \cdots, A_{\delta(f)})}\, f_\rightarrow R$$

Figure 3: Sequent rules for **LPR**

The equivalence of the axiomatic with the sequent presentation, as well as Cut elimination for all instances of **LPR** can be proven by arguments that are entirely parallel to the corresponding proofs for **NL**.

The results from the previous section on **NL**$\diamond$ carry over to all instances of **LPR** without further ado.

Lemma 5 *Let $X[Y] \Rightarrow A$ be a theorem of **LPR**. Then there is a type B such that*

1. ***LPR*** $\vdash Y \Rightarrow B$
2. ***LPR*** $\vdash X[B] \Rightarrow A$
3. *There is a type occurring in $X[Y] \Rightarrow A$ which contains at least as many connectives as B.*

Proof: Parallel to the proof of lemma 2. ⊣

So there is a finite axiomatization of any fragment of an **LPR** that has an upper limit for the complexity of the types involved. The notions of an **LPR**-grammar and of the language recognized by such a grammar can be adapted from the corresponding notions related to **NL**$\diamond$ in an obvious way. Finite axiomatizability thus amounts to the fact that every **LPR**-grammar is equivalent to some context free grammar. On the other hand, any **NL**-grammar is an **LPR**-grammar, and since any context free language is recognized by some **NL**-grammar, we obtain the following generalization of theorem 1:

Theorem 2 ***LPR**-grammars recognize exactly the class of context free languages.*

Proof: Analogous to the corresponding proof for **NL**$\diamond$. ⊣

Up to now we only considered calculi of pure residuation, i.e. calculi that do without any structural rules. One might wonder what impact the presence of structural rules has on the generative capacity. We will consider the structural rules Associativity (A), Permutation (P), Contraction (C), Expansion (E) and Weakening (W) in the sequel, as applying to some distinguished binary mode f. For simplicity, we write $\circ$ instead of $f_\circ$ below. The sequent versions of these rules are given in figure 4. (The double line in A indicates that these are actually two rules, left associativity and right associativity.)

$$\frac{X[Y \circ (Z \circ W)] \Rightarrow A}{X[(Y \circ Z) \circ W] \Rightarrow A}\,A \qquad \frac{X[Y \circ Z] \Rightarrow A}{X[Z \circ Y] \Rightarrow A}\,P$$

$$\frac{X[Y \circ Y] \Rightarrow A}{X[Y] \Rightarrow A}\,C \qquad \frac{X[Y] \Rightarrow A}{X[Y \circ Y] \Rightarrow A}\,E$$

$$\frac{X[Y] \Rightarrow A}{X[Y \circ Z] \Rightarrow A}\,W$$

Figure 4: Structural Rules

Would the proof for lemma 5 still go through if we add some of these rules to the calculus? Certainly not for A. If the induction hypothesis would hold for the sequent on top, this would not guarantee that there is a witness for $Y \circ Z$, and likewise for the opposite direction. Likewise, the induction step would not work for W, because the induction hypothesis—the lemma holds for the premise sequent—does not guarantee that there is a witness for Z. It doesn't work for C either because we cannot be sure whether the witness for the two occurrences of Y in the premise are identical. If they aren't, contraction cannot be applied anymore after replacing the two Ys by their witnesses. However, P and E are well-behaved.

Lemma 6 *Let* $\mathbf{C}$ *be some multimodal calculus that comprises the sequent rules for* ***LPR*** *and a subset of the structural rules* $\{P, E\}$ *for each binary mode of* $\mathbf{C}$. *Let* $X[Y] \Rightarrow A$ *be a theorem of* $\mathbf{C}$. *Then there is a type* B *such that*

1. $\mathbf{C} \vdash Y \Rightarrow B$
2. $\mathbf{C} \vdash X[B] \Rightarrow A$
3. *There is a type occurring in* $X[Y] \Rightarrow A$ *which contains at least as many connectives as* B.

Proof: By induction over sequent derivations. For the logical rules, the induction step was established above. As for P, the witness of any substructure of X in the conclusion is identical to the corresponding witness in the premise, and likewise for substructures of Y and Z. As for E, the witness for a substructure of X in the conclusion is inherited from the premise. Suppose Y has a substructure Z, and we want to know whether there is a witness for the occurrence of Z in the left occurrence of Y in the conclusion. By hypothesis, we know that for some type B with a complexity that is not more complex than the most complex type in the premise sequent, it holds that $\mathbf{C} \vdash X[Y[B]] \Rightarrow A$, and $\mathbf{C} \vdash B \Rightarrow Z$. By applying E, we obtain $\mathbf{C} \vdash X[Y[B] \circ Y[B]] \Rightarrow A$, and by Cut we get $\mathbf{C} \vdash X[Y[B]] \circ Y[Z] \Rightarrow A$. ⊣

Thus we have

Theorem 3 *A type logical grammar that is based on a calculus which extends a version of* ***LPR*** *with* P *or* E *for some of its binary modes recognizes only context free languages.*

Proof: Immediate. ⊣

Let me conclude this section with some remarks on the relation of my results to (Kandulski, 1995) and (Kandulski, 2002). The central theorem of the latter work is almost identical to my theorem 2 (and my theorem 1 is just a corollary of this). Kandulski gives an axiomatization for **LPR**using Cut as the only rule, and he shows that derivations in this axiomatic calculus can be normalized in such a way that only finitely instances of the axioms are relevant for a given **LPR**-grammar. There is a minor difference between his results and mine: Kandulski requires

that the set of designated types of a type logical grammar is a singleton containing only one atomic type. The restriction to atomic types is in fact essential for his proof to go through. The same holds *ceteris paribus* for the context freeness proof for **NL** in (Kandulski, 1988). In this respect the results from the present paper are somewhat more general. Furthermore, (Kandulski, 1995) presents a proof that grammars based on **NL**+P only recognize context free languages. Again, the proof makes essential use of the restriction to atomic designated types.

Even though the results obtained in the present paper have an considerable overlap with Kandulski's prior work, the proof strategy used here is novel, and arguably simpler. It is also easier to generalize, as the application to Expansion illustrates.

5. Conclusion

In this paper a new strategy for proving the context freeness of a class of type logical grammars was proposed. The basic idea for the construction of context free grammars from type logical grammars is adapted from (Pentus, 1993). The proof of the correctness of the construction is different (and much simpler) though; it is based on a property of sequent derivations that can be seen as a variant of Roorda's (1991) interpolation lemma. It was shown that this property is shared by all multimodal logics of pure residuation, i.e. any pure or mixed calculi using only families of residuated operators of arbitrary arity. Prominent instances of this family of logics are **NL** and **NL**$\diamond$. It was furthermore proved that this property of sequent calculi is preserved by adding the structural rules of Permutation or Expansion. Any categorial grammar based on one of these logics recognizes a context free language. Conversely, by a slight variation of Cohen's (1967) argument for Ł it can be shown that any context free language is recognized by some **LPR** comprising at least one binary product, but no structural rules.

Further work is required to gain a deeper understanding on the impact of structural rules on generative capacity. The proof strategy that was proposed in this paper can be used as a recipe to establish context freeness for extensions of **LPR** with certain structural postulates, including interaction postulates involving several modes. However, it is not always applicable, as the example of the associative Lambek calculus demonstrates. So it would be desirable to identify sufficient conditions when structural rules preserve context freeness.

References

Bar-Hillel, Yehoshua. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29:47–58.

Bar-Hillel, Yehoshua, C. Gaifman and E. Shamir. 1960. On Categorial and Phrase Structure Grammars. *Bulletin of the Research Council of Israel*, F(9):1–16.

Buszkowski, Wojciech. 1986. Generative Capacity of Nonassociative Lambek Calculus. *Bulletin of the Polish Academy of Sciences: Mathematics*, 34:507–518.

Carpenter, Bob. 1999. The Turing-completeness of multimodal categorial grammars. Papers presented to Johan van Benthem in honor of his 50th birthday. European Summer School in Logic, Language and Information, Utrecht.

Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton.

Cohen, Joel M. 1967. The equivalence of two concepts of Categorial Grammar. *Information and Control*, 10:475–484.

Jäger, Gerhard. 2001. On the Generative Capacity of Multimodal Categorial Grammars. to appear in *Journal of Language and Computation*.

Kandulski, Maciej. 1988. The equivalence of nonassociative Lambek categorial grammars and context-free grammars. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 34:41–52.

Kandulski, Maciej. 1995. On commutative and nonassociative syntactic calculi and categorial grammars. *Mathematical Logic Quarterly*, 41:217–135.

Kandulski, Maciej. 2002. On Generalized Ajdukiewicz and Lambek Calculi and Grammars. manuscript, Poznan University.

Lambek, Joachim. 1958. The Mathematics of Sentence Structure. *American Mathematical Monthly*, 65:154–170.

Lambek, Joachim. 1961. On the Calculus of Syntactic Types. In Roman Jakobson, editor, *Structure of Language and Its Mathematical Aspects*. Providence, RI.

Moortgat, Michael. 1988. *Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus*. Dordrecht: Foris.

Moortgat, Michael. 1996. Multimodal linguistic inference. *Journal of Logic, Language and Information*, 5(3/4):349–385.

Morrill, Glyn. 1990. Intensionality and Boundedness. *Linguistics and Philosophy*, 13:699–726.

Pentus, Martin. 1993. Lambek grammars are context-free. In *Proceedings of the 8th Annual IEEE Symposium on Logic in Computer Science*. Montreal.

Roorda, Dirk. 1991. *Resource logics: Proof-theoretical investigations*. Ph.D. thesis, University of Amsterdam.

Shieber, Stuart. 1985. Evidence against the non-context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.