Extending Lambek Grammars to Basic Categorial Grammars

Wojciech Buszkowski
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań Poland

Journal of Logic, Language and Information 5.3-4: 279–295, 1996

Abstract

PENTUS [24] proves the equivalence of LCG's and CFG's, and CFG's are equivalent to BCG's by the Gaifman theorem [1]. This paper provides a procedure to extend any LCG to an equivalent BCG by affixing new types to the lexicon; a procedure of that kind was proposed as early, as COHEN [12], but it was deficient [4]. We use a modification of PENTUS' proof and a new proof of the Gaifman theorem on the basis of the Lambek calculus.

1 Introduction and preliminaries

A *categorial grammar* is a quadruple $G = (V_G, I_G, s_G, R_G)$, such that V_G is a nonempty finite lexicon (alphabet), I_G is a mapping which assigns a finite set of types to each atom $v \in V_G$, s_G is a designated atomic type, and R_G is a type change system. One refers to V_G, I_G, s_G and R_G as *the lexicon, the initial type assignment, the principal type* and *the system* of G. We say that G assigns type a to string $v_1 \dots v_n$ ($v_i \in V_G$), if the sequent $a_1 \dots a_n \to a$ is derivable in R_G, for some $a_i \in I_G(v_i)$, $i = 1, \dots, n$. The set $L(G)$, called *the language* of G, consists of all the strings on V_G which are assigned type s_G by G. Two grammars are said to be *equivalent,* if they yield the same language.

Types are formed out of atomic types by means of operation symbols $\backslash, /, \star$, called *left residuation, right residuation,* and *product,* respectively. We denote types by a, b, c, atomic types by p, q, r, and finite strings of types by X, Y, Z. *Basic Categorial Grammars* (BCG's) admit the system **B** whose formulas are product-free sequents $a_1 \dots a_n \to a$, and its axioms and rules are as follows:

(Ax) $a \to a$

($\backslash$1) $XaZ \to c$, $Y \to b \vdash XY(b\backslash a)Z \to c$

(/1) $XaZ \to c$, $Y \to b \vdash X(a/b)YZ \to c$.

The sequent $X \to a$ is derivable in **B** if, and only if, X reduces to a by the standard reduction procedure based on the rules:

(R\) $b(b\backslash a) \Rightarrow a$

(R/) $(a/b)b \Rightarrow a$,

and consequently, BCG's are precisely categorial grammars in the sense of [1].

Lambek Categorial Grammars (LCG's) are based on the system **L** equal to **B** enriched with additional rules:

(\2) $bX \to a \vdash X \to b\backslash a$

(/2) $Xb \to a \vdash X \to a/b$,

where X is nonempty (dropping this constraint yields a stronger system **L1**).

The full *Lambek Calculus* admits types with product and can be axiomatized by (Ax), (\1), (/1), (\2), (/2) together with: the following product introduction rules:

($\star$1) $XabY \to c \vdash X(a \star b)Y \to c$

($\star$2) $X \to a$, $Y \to b \vdash XY \to a \star b$.

The latter system is denoted **LP**, and **LP1** is defined in an obvious way. In each of the four variants of the Lambek Calculus, axioms (Ax) can be restricted to atomic types a. They are Gentzen-style systems without structural rules [13]. All the systems mentioned above are decidable and closed under *the cut rule*:

(CUT) $XaZ \to b$, $Y \to a \vdash XYZ \to b$,

which has been proved in LAMBEK [20] for **LP**.

The Lambek Calculus and its sub- and supersystems are closely related to several issues of current interest in logic, as e.g. linear logics [17], action logic [25], gaggle theory [14], labelled deductive systems [16], and in natural language semantics and computational linguistics (see VAN BENTHEM [2], MOORTGAT [22]). A thorough logical discussion of the domain can be found in [10, 5], and many linguistic applications in [23].

In terms of type theory, **B** is a purely applicative system, while Lambek systems employ lambda-abstraction. A problem which has quite early appeared in this discipline is whether lambda-abstraction affects generative capacity. Strictly, the question is whether LCG's are equivalent to BCG's. In [1], BCG's are proven to be equivalent to Context-Free Grammars (CFG's) (the Gaifman theorem), and the authors conjecture the same equivalence holds for LCG's. This conjecture, repeated in CHOMSKY [11], was referred to as the Chomsky conjecture. COHEN [12] shows that each BCG is equivalent to some LCG, but his proof of the converse statement contains essential errors [4]. There have been obtained partial results in this direction [4, 7, 9]; for the Nonassociative Lambek Calculus even a strong equivalence with BCG's is given in [6, 19]. Finally PENTUS [24] proves the conjecture for the full calculus **LP**. It follows

from his theorem that each LCG is equivalent to some CFG, hence to some BCG, and the same holds for categorial grammars based on **L1**, **LP**, **LP1**. (No kind of strong equivalence is possible here, since Lambek systems are structurally complete [7].)

In this paper we prove that each LCG can be transformed into an equivalent BCG in a natural way: namely one expands the initial type assignment of the LCG by affixing new types b, such that there is type a in the initial type assignment with $a \to b$ derivable in **L**. This is precisely the way COHEN has tried in his deficient proof, but our expansion is more subtle and essentially uses the methods elaborated by PENTUS as well, as a proof of the Gaifman theorem on the basis of **L**. From the linguistic point of view, such a natural extension of a grammar to another one is quite desirable, since new types have a clear linguistic meaning. On the other hand, going directly the way of PENTUS yields an artificial grammar: for the given LCG one constructs an equivalent CFG and, then, transforms the latter into an equivalent BCG using the construction from the Gaifman theorem; eventually, there are no semantic connections between the initial type assignment of the third grammar and that of the first grammar.

The paper consists of four sections. Section 2 provides a proof of the Gaifman theorem which essentially employs the logic of **L**. Section 3 adapts the Pentus theorem to product-free types. The main result and some final comments are given in section 4.

2 The Gaifman theorem proven from L

Recall that a CFG is a quadruple $\Gamma = (V_\Gamma, N_\Gamma, s_\Gamma, R_\Gamma)$ such that V_Γ is a nonempty finite set of *terminal* symbols, N_Γ is a nonempty finite set of *nonterminal* symbols which is disjoint with V_Γ, $s_\Gamma \in N_\Gamma$ is *the initial symbol*, and R_Γ is a finite set of *production rules* of the form:

(R1) $p \Rightarrow p_1 \dots p_n$, where $p, p_i \in N_\Gamma$,

(R2) $p \Rightarrow v$, where $p \in N_\Gamma$, $v \in V_\Gamma$.

Nonterminal symbols of a CFG are symbolized by the same letters as atomic types, since both kinds of symbols will be identified in the sequel. The relation $p \Rightarrow_\Gamma X$, where $p \in N_\Gamma$, $X \in N_\Gamma^+$, is recursively defined as follows:

(D1) $p \Rightarrow_\Gamma p$, for all $p \in N_\Gamma$,

(D2) if $p \Rightarrow p_1 \dots p_n$ is a rule (R1) from R_Γ and $p_i \Rightarrow_\Gamma X_i$, for all $i = 1, \dots, n$, then $p \Rightarrow_\Gamma X_1 \dots X_n$.

The language of Γ is the set $L(\Gamma)$ which consists of all strings $v_1 \dots v_n$, $n \geq 1$, such that, for some nonterminal symbols $p_1, \dots, p_n$ with $p_i \Rightarrow v_i \in R_\Gamma$, $i = 1, \dots, n$, there holds $s_\Gamma \Rightarrow_\Gamma p_1 \dots p_n$. The CFG Γ_1 is *equivalent* to the CFG Γ_2 if $L(\Gamma_1) = L(\Gamma_2)$. It is well known that each CFG is equivalent to some CFG in the Chomsky Normal Form; the latter's rules (R1) are always of the form $p \Rightarrow p_1 p_2$ [18].

The CFG Γ is *equivalent* to the BCG G if $L(\Gamma) = L(G)$. The Gaifman theorem [1] establishes the equivalence of BCG's and CFG's. It can be formulated as the conjunction of the following statements:

(I) each BCG is equivalent to some CFG,

(II) each CFG is equivalent to some BCG whose initial type assignment uses at most types of the form $p, p/q, (p/q)/r$, where p, q, r are atomic.

Statement (I) is obvious: rules (R$\backslash$) and (R/) decrease the complexity of types, and consequently, the given BCG G is equivalent to the CFG Γ with $V_\Gamma = V_G$, $N_\Gamma =$ the set of subtypes of the types from I_G, $s_\Gamma = s_G$, and R_Γ consisting of all rules:

$$a \Rightarrow (a/b)b;\ b \Rightarrow a(a\backslash b),$$

for $a, b \in N_\Gamma$.

Statement (II) is nontrivial; actually, it is equivalent to the Greibach Normal Form theorem in the theory of CFG's [18]. In this section we give a proof of (II) with the aid of the Lambek Calculus; the idea of this proof has already been announced in [8, 7]. In [8] there is given another proof of (II) which relies upon congruences and transformations in the algebra of phrase structures.

We use axiomatic extensions of **L**, first studied in [3]. Let R be a set of product-free sequents $X \to a$ ($X \neq \Lambda$). (Λ stands for the empty string). $\mathbf{L}(R)$ denotes the system axiomatized by (Ax) and all the sequents from R (as new axioms) together with the inference rules of **L** and (CUT). An equivalent Gentzen style axiomatization can be given as follows. First, observe each sequent is deductively equivalent to a sequent of the form $X \to p$ (p is atomic) on the basis of **L**. So, we may assume each sequent in R is of the latter form. The system $\mathbf{GL}(R)$ is axiomatized by (Ax), (/1), ($\backslash$1), (/2), ($\backslash$2) and the special rules:

$$\text{(SR) } X_1 \to a_1, \ldots, X_n \to a_n \vdash X_1 \ldots X_n \to p,$$

for all sequents $a_1 \ldots a_n \to p$ in R.

Lemma 1 *GL(R) is closed under (CUT).*

Proof. The proof proceeds by triple induction: (1) on the complexity of type a in (CUT), (2) on the derivation of the first premise, (3) on the derivation of the second premise. The crucial point is that the conclusion of (SR) cannot be the second premise of (CUT), if a in the first premise is the type designated in (/1) or ($\backslash$1). □

Lemma 2 $\mathbf{GL}(R)$ *and* $\mathbf{L}(R)$ *yield the same derivable sequents.*

Proof. Using (CUT), we show that $\mathbf{L}(R)$ is closed under each rule of $\mathbf{GL}(R)$. Conversely, each sequent from R is derivable in $\mathbf{GL}(R)$, by (Ax) and (SR), and $\mathbf{GL}(R)$ is closed under (CUT). Then, $\mathbf{L}(R)$ and $\mathbf{GL}(R)$ are equivalent. □

Let us note that lemma 2 does not imply the decidability of systems $\mathbf{L}(R)$, even for finite sets R (rules (SR) can forget information). As shown in [3], each recursively enumerable language can be generated by a categorial grammar based on some system $\mathbf{L}(R)$ with R finite.

We are concerned with especially simple sets R which consist of finitely many sequents of the form:

$$\text{(S) } p_1 \dots p_n \to p;$$

these sequents are naturally related to production rules (R1). For those sets R, systems $\mathbf{L}(R)$ are decidable [3]. By R^Γ we denote the set of all sequents (S) related to the production rules (R1) of the CFG Γ.

Lemma 3 *For any $p, p_1, \dots, p_n \in N_\Gamma$, $p \Rightarrow_\Gamma p_1 \dots p_n$ if, and only if, $\mathbf{L}(R^\Gamma) \vdash p_1 \dots p_n \to p$.*

Proof. Since $\mathbf{L}(R^\Gamma)$ admits (CUT), then 'only if' holds. For 'if', it is enough to notice that each derivation of $p_1 \dots p_n \to p$ in $\mathbf{GL}(R^\Gamma)$ uses at most (Ax) and (SR), hence it amounts to a derivation in Γ (up to the direction of arrows). □

Now, with each CFG Γ we associate a categorial grammar $G(\Gamma)$ whose system is $\mathbf{L}(R^\Gamma)$ and other components are defined as follows: $V_{G(\Gamma)} = V_\Gamma$, $s_{G(\Gamma)} = s_\Gamma$ and $I_{G(\Gamma)}(v)$ consists of all nonterminal symbols p such that (R2) belongs to R_Γ. As an immediate consequence of lemma 3, we obtain:

Fact 1 $L(G(\Gamma)) = L(\Gamma)$.

The BCG G is said to be *derivable* from the CFG Γ, if the lexicon and the principal type of G are those of $G(\Gamma)$, and the initial type assignment of G fulfills the condition:

(DER) if $a \in I_G(v)$, then $\mathbf{L}(R^\Gamma) \vdash p \to v$, for some $p \in I_{G(\Gamma)}(v)$.

If G is derivable from Γ, then $L(G) \subset L(G(\Gamma))$ (since $\mathbf{L}(R^\Gamma)$ admits (CUT) and is stronger than $\mathbf{B}$). So, by fact 1, we obtain:

Lemma 4 *If a BCG G is derivable from a CFG Γ, then $L(G) \subset L(\Gamma)$.*

Accordingly, in order to find a BCG equivalent to the given CFG Γ, it suffices to construct a BCG G derivable from Γ and such that $L(\Gamma) \subset L(G)$. To accomplish this goal we need the following properties of the Lambek Calculus:

(L1) if $\mathbf{L}(R) \vdash qr \to p$, then $\mathbf{L}(R) \vdash r \to q\backslash p$,

(L2) $\mathbf{L} \vdash q\backslash p \to (q\backslash t)/(p\backslash t)$,

(L3) $\mathbf{L} \vdash q \to p/(q\backslash p)$,

(L4) if $\mathbf{L}(R) \vdash a \to b$, then $\mathbf{L}(R) \vdash a/c \to b/c$,

for all (not necessarily atomic) types p, q, r, t, a, b, c. (L1) holds, by ($\backslash 2$). (L2) follows from $\mathbf{L} \vdash q(q\backslash p)(p\backslash t) \to t$, by ($\backslash 2$) and ($/2$). (L3) is a consequence of $\mathbf{L} \vdash p(p\backslash q) \to q$, by ($/2$). For (L4), $(a/c)c \to a$ is derivable in $\mathbf{L}$, hence $a \to b$ entails $(a/c)c \to b$, by (CUT), which yields $a/c \to b/c$, by ($/2$).

Let Γ be a CFG in the Chomsky Normal Form. We define a mapping I which assigns a finite set of types to each nonterminal symbol of Γ and satisfies the condition:

(DER') if $a \in I(p)$, then $\mathbf{L}(R^\Gamma) \vdash p \to a$.

We set $I(p) = I_1(p) \cup I_2(p)$, where I_1, I_2 are defined as follows. For any production rule $p \Rightarrow qr$ from R_Γ we put types:

$$q\backslash p \text{ and } (q\backslash t)/(p\backslash t), \text{ for all } t \in N_\Gamma, \tag{1}$$

into $I_1(r)$. Additionally, we also put s_Γ into $I_1(s_\Gamma)$. Further, for all types a, p, q, if $a \in I_1(p)$, then we put type:

$$a/(q\backslash p) \tag{2}$$

into $I_2(q)$. This finishes the construction of I. (DER') is an easy consequence of (L1)-(L4). We only show the case (2). Since $\mathbf{L}(R^\Gamma) \vdash p \to a$, by (DER') and the fact that $a \in I_1(p)$, then:

$$\mathbf{L}(R^\Gamma) \vdash p/(q\backslash p) \to a/(q\backslash p),$$

by (L4), and consequently, $\mathbf{L}(R^\Gamma) \vdash q \to a/(q\backslash p)$, by (L3) and (CUT).

We define a BCG G derivable from Γ by setting:

$$I_G(v) = \bigcup\{I(p) : p \Rightarrow v \in R_\Gamma\}, \tag{3}$$

for $v \in V_G = V_\Gamma$.

We must show $L(\Gamma) \subset L(G)$. We need simple properties of derivations in Γ. (D2) for Γ takes the form:

(D2') if $q \Rightarrow_\Gamma X$ and $r \Rightarrow_\Gamma Y$, then $p \Rightarrow_\Gamma XY$,

for any production rule $p \Rightarrow qr$ from R_Γ. A derivation in Γ is said to be *regular*, if $Y = r$ for each application of (D2'). (Clearly, regular derivations can be simulated by finite state acceptors.) The next lemma exhibits regular subderivations of each derivation in Γ.

Lemma 5 *If $p \Rightarrow_\Gamma qX$, then there are a number $k \geq 0$, nonterminal symbols $q_1, \ldots, q_k$, and strings $X_1, \ldots, X_k$ such that $X = X_1 \ldots X_k$, $q_i \Rightarrow_\Gamma X_i$, for all $i = 1, \ldots, k$, and $p \Rightarrow_\Gamma qq_1 \ldots q_k$ has a regular derivation.*

Proof. Induction on the length of X. For $X = \Lambda$, we have $p = q$ and $k = 0$. Assume $X \neq \Lambda$. Then, there exist a production rule $p \Rightarrow rs$ and strings Y, Z such that $X = YZ$ and $r \Rightarrow_\Gamma qY$, $s \Rightarrow_\Gamma Z$. Since $Z \neq \Lambda$, then Y is shorter than X. By the induction hypothesis, there are $k \geq 0$, $q_1, \ldots, q_k$ and $X_1, \ldots, X_k$ such that $Y = X_1 \ldots X_k$, $q_i \Rightarrow_\Gamma X_i$, for all $i = 1, \ldots, k$, and $r \Rightarrow_\Gamma qq_1 \ldots q_k$ has a regular derivation. We take $q_{k+1} = s$ and $X_{k+1} = Z$. $\square$

Lemma 6 *Let $p \Rightarrow_\Gamma qq_1 \dots q_k$ $(k \geq 0)$ have a regular derivation. Then, for any type $a \in I_1(p)$, there are types $b \in I(q)$ and $b_i \in I_1(q_i)$, for $i = 1, \dots, k$, such that $\mathbf{B} \vdash bb_1 \dots b_k \to a$ and rule $(\backslash 1)$ (equivalently: (R\)) is not applied in the latter derivation.*

Proof. For $k = 0$, we take $b = a$. Assume $k \geq 2$. The regular derivation proceeds by the following sequence of production rules:

$$p \Rightarrow r_k q_k,\ r_k \Rightarrow r_{k-1}q_{k-1}, \dots, r_3 \Rightarrow r_2 q_2,\ r_2 \Rightarrow qq_1, \tag{4}$$

for some nonterminal symbols $r_2, \dots, r_k$. By (1), I_1 satisfies the condition:

$$r_k\backslash p \in I_1(q_k),\ (r_{k-1}\backslash p)/(r_k\backslash p) \in I_1(q_{k-1}), \dots, (q\backslash p)/(r_2\backslash p) \in I_1(q_1); \tag{5}$$

the left-hand types are denoted $b_k, \dots, b_1$, respectively. Evidently:

$$\mathbf{B} \vdash b_1 \dots b_k \to q\backslash p,$$

and (R\) is not applied in this derivation. Now, choose $a \in I_1(p)$. By (2), $a/(q\backslash p) \in I_2(q)$, which yields the thesis with $b = a/(q\backslash p)$. Case $k = 1$ is particular: $p \Rightarrow qq_1$ is the only rule in (4), and we set $b_1 = q\backslash p$ and $b = a/(q\backslash p)$. □

Accordingly, the BCG constructed above can simulate regular derivations in Γ. We show it can simulate arbitrary derivations in Γ.

Lemma 7 *Assume $p \Rightarrow_\Gamma p_1 \dots p_n$. Then, for any $a \in I_1(p)$, there are types $c_i \in I(p_i)$, for $i = 1, \dots, n$, such that $\mathbf{B} \vdash c_1 \dots c_n \to a$, and rule (R\) is not applied in the latter derivation.*

Proof. Induction on n. For $n = 1$, the derivation is regular, hence lemma 6 yields the thesis. Assume $n > 1$. By lemma 5, there are $k, q_1, \dots, q_k$ and $X_1, \dots, X_k$ such that $p_2 \dots p_n = X_1 \dots X_k$ (hence $k \neq 0$), $q_i \Rightarrow_\Gamma X_i$, for $i = 1, \dots, k$, and $p \Rightarrow_\Gamma p_1 q_1 \dots q_k$ has a regular derivation. Choose $a \in I_1(p)$. By lemma 6, there are types $c_1 \in I(p_1)$ and $b_i \in I_1(q_i)$, $i = 1, \dots, k$, such that:

$$\mathbf{B} \vdash c_1 b_1 \dots b_k \to a$$

without $(\backslash 1)$. By the induction hypothesis, since $b_i \in I_1(q_i)$ and $q_i \Rightarrow_\Gamma X_i$, then we find a string Y_i of types assigned by I to the corresponding symbols from X_i, such that $\mathbf{B} \vdash Y_i \to b_i$ without $(\backslash 1)$. Consequently,

$$\mathbf{B} \vdash c_1 Y_1 \dots Y_k \to a$$

holds by (CUT), and we set $c_2 \dots c_n = Y_1 \dots Y_k$. Clearly, $(\backslash 1)$ is not applied. □

Fact 2 *Let Γ be a CFG in the Chomsky Normal Form, and let G be the BCG derivable from Γ, constructed according to (3). Then, $L(\Gamma) = L(G)$.*

PROOF. Lemma 4 yields $L(G) \subset L(\Gamma)$. For the converse inclusion, assume $v_1 \dots v_n \in L(\Gamma)$. Then, there are $p_i \in N_\Gamma$, $i = 1, \dots, n$, such that $p_i \Rightarrow v_i \in R_\Gamma$ and $s_\Gamma \Rightarrow_\Gamma p_1 \dots p_n$. Since $s_\Gamma \in I_1(s_\Gamma)$, then, by lemma 7, there are types $c_i \in I(p_i)$, $i = 1, \dots, n$, such that $\mathbf{B} \vdash c_1 \dots c_n \to s_\Gamma$ (without (\1)). By (3), $c_i \in I_G(v_i)$, $i = 1, \dots, n$, which yields $v_1 \dots v_n \in L(G)$. □

The proof of fact 2 shows that part (II) of the Gaifman theorem holds true. Since rule (\1) is not used, one can drop it from **B**. Then, of course, both lemma 4 and fact 2 remain true. Now, if **B** lacks (\1), then types of the form $a\backslash b$ are treated as atomic types, actually. In the definition of G, we replace each type $p\backslash q$ by the atomic type p^q. So, types in (1) are changed into:

$$p^q,\ p^t/q^t,$$

and types in (2) are changed into types a/q^p, where a are of the above form. Consequently, a BCG G equivalent to the CFG Γ can be constructed with types in I_G restricted to the three forms in (II). However, this transformation hiddens the fact that types in I_G are derived from nonterminal symbols of Γ by means of **L**, and this possibility has been our major concern in this section. Due to it, the CFG equivalent to the given LCG, to be constructed in the next section with applying Pentus' methods, can be transformed into an equivalent BCG by an **L**-derivable expansion of the initial type assignment of the LCG (see section 4).

3 Interpolation and binary reductions for L

By $\rho(a)$ we denote *the complexity of type* a, i.e. the number of occurrences of atomic types in a. We also set:

$$\rho(a_1 \dots a_n) = \rho(a_1) + \dots + \rho(a_n),\ \rho(X \to a) = \rho(X) + \rho(a).$$

By $l(X)$ we denote the length of string X. For a set P, of atomic types, $TP_n(P)$ denotes the set of all types a such that $\rho(a) \leq n$ and all atomic types occurring in a are in P. $Tp_n(P)$ stands for the set of all product-free types in $TP_n(P)$.

The major lemma in PENTUS [24] is the following *binary reduction lemma* (the BR-lemma):

- *For any set P and any number $n \geq 1$, if $\mathbf{LP} \vdash X \to a$, $X \in TP_n(P)$, $l(X) \geq 2$, $a \in TP_n(P)$, then there exist types $b, c, d \in TP_n(P)$ and strings Y, Z such that $X = YbcZ$, $\mathbf{LP} \vdash bc \to d$ and $\mathbf{LP} \vdash YdZ \to a$.*

The BR-lemma has earlier been proven in [4, 9] for some special families of product-free types, while [24] succeeds in establishing it for arbitrary types.

It immediately follows from the BR-lemma that each LCG (with product) is equivalent to some CFG. Fix an LCG G (with product). We choose a positive integer n and a finite set P such that $TP_n(P)$ contains all types appearing in I_G. The CFG Γ is defined as follows: $V_\Gamma = V_G$, $N_\Gamma = TP_n(P)$, $s_\Gamma = s_G$, and R_Γ consists of production rules:

(G1) $d \Rightarrow bc$, for all $b, c, d \in N_\Gamma$ such that $\mathbf{LP} \vdash bc \to d$,

(G1') $b \Rightarrow a$, for all $a, b \in N_\Gamma$ such that $\mathbf{LP} \vdash a \to b$,

(G2) $a \Rightarrow v$, for all $v \in V_G$, $a \in I_G(v)$.

The BR-lemma yields $L(G) \subset L(\Gamma)$, and the converse inclusion also holds, since **LP** is closed under (CUT). Consequently, G is equivalent to Γ.

The aim of this paper is to transform the given LCG (without product) into an equivalent BCG by an **L**-derivable expansion of the initial type assignment of the LCG. Since **L** is a conservative subsystem of **LP**, then we can use the above construction to find a CFG Γ, equivalent to the given LCG G. Applying the construction from section 2, Γ can be transformed into a BCG G', derivable from Γ and equivalent to Γ. One easily checks $I_{G'}$ results from expanding I_G by means of **L** (see section 4).

The failure of this transformation is that it, actually, yields a pseudo-BCG G' in which the product symbol can appear in types from $I_{G'}$. For it does not follow from the Pentus proof of the BR-lemma that, if X consists of product-free types, then there exist types b, c, d such that d is product-free and the remaining conditions hold. Pseudo-BCG's are formally and linguistically ugly: the logic of **B** does not touch the product, hence types of the form $a \star b$ are treated as atomic types, and their compound structure is an overcomplication. Within the product-free world, which is the world of most linguistic examples of categorial grammars, we would prefer to transform the given LCG into a normal, i.e. product-free, BCG. The product-free transformation is also desirable for semantic reasons: the standard typed lambda calculus suffices to transform the semantic denotations of lexical units, corresponding to the initial LCG, into those corresponding to the resulting BCG, while with product appearing in types one must use an extended lambda calculus [2, 21].

To accomplish this goal we need the BR-lemma for product-free types, which will be proven below. The proof follows the Pentus proof rather closely, but an essential change must be done in *the interpolation lemma*, established for **LP** in Roorda [26].

By $\rho(p, a)$ we denote the number of occurrences of the atomic type p in type a, and $\rho(p, X)$, $\rho(p, X \to a)$ are defined as $\rho(X)$, $\rho(X \to a)$. Let $\mathbf{LP} \vdash XYZ \to a$ with $Y \neq \Lambda$. The type y is called *an interpolant* of string Y in the latter context, if the following conditions are satisfied:

(I1) $\mathbf{LP} \vdash Y \to y$ and $\mathbf{LP} \vdash XyZ \to a$,

(I2) $\rho(p, y) \leq \min(\rho(p, Y), \rho(p, XZ \to a))$, for every atomic type p.

As shown in [26], interpolants exist for all strings $Y \neq \Lambda$ in any context $\mathbf{LP} \vdash XYZ \to a$. The Pentus proof of the BR-lemma relies on this interpolation property: the type d is chosen as an interpolant of an interval bc in $\mathbf{LP} \vdash YbcZ \to a$.

For the case of **L**, the Roorda interpolation property does not hold. Consider the context:

$$\mathbf{L} \vdash pqr \to (s/pqr)\backslash s. \tag{6}$$

Here we write s/pqr for $((s/r)/q)/p$. In general, we define the abbreviated notation a/X and $X\backslash a$ by induction on $l(X)$:

(N1) $a/\Lambda = \Lambda\backslash a = a$,

(N2) $a/(Xb) = (a/b)/X$, $(Xb)\backslash a = b\backslash(X\backslash a)$.

Clearly, (6) holds, by (Ax), (/1) and (\2). We show that $q \star r$ is the only interpolant y of string qr in this context. (Consequently, there is no product-free interpolant!) By (I2), $\rho(q, y) \leq 1$, $\rho(r, y) \leq 1$ and $\rho(t, y) = 0$, for any atomic type t different from q and r. So, the only possible candidates for y are:

$$q,\ r,\ q/r,\ q\backslash r,\ r/q,\ r\backslash q,\ q \star r,\ r \star q,$$

and only $y = q \star r$ satisfies $\mathbf{LP} \vdash qr \to y$.

Nevertheless, we obtain an interpolation property for $\mathbf{L}$ with a modified notion of an interpolant. By *an interpolant of string* $Y \neq \Lambda$ *in the context* $\mathbf{L} \vdash XYZ \vdash a$ we mean a string $y_1 \dots y_n$ $(n > 0)$, of product-free types, such that there are nonempty strings $Y_1, \dots, Y_n$ satisfying $Y = Y_1 \dots Y_n$ and the following conditions:

(LI1) $\mathbf{L} \vdash Y_i \to y_i$, for $i = 1, \dots, n$,

(LI2) $\mathbf{L} \vdash Xy_1 \dots y_n Z \to a$,

(LI3) $\rho(p, y_i) \leq \min(\rho(p, Y_i), \rho(p, XY'Z \to a))$, $Y' = Y_1 \dots Y_{i-1} Y_{i+1} \dots Y_n$, for $i = 1, \dots, n$,

(LI4) $\rho(p, y_1 \dots y_n) \leq \min(\rho(p, Y), \rho(p, XZ \to a))$,

for all atomic types p. That means, each type y_i is an interpolant of the corresponding string Y_i, and type $y_1 \star \dots \star y_n$ is an interpolant of string Y in the previous sense. We sketch the proof of the interpolation lemma for $\mathbf{L}$:

Lemma 8 *If* $\mathbf{L} \vdash XYZ \to a$, $Y \neq \Lambda$, *then there is an interpolant of string* Y *in this context.*

Proof. We proceed by induction on derivations of $XYZ \to a$ in $\mathbf{L}$.

If $XYZ \to a$ is (Ax), then $Y = a$, $XZ = \Lambda$, and $y = a$ is an interpolant of y. Rules (/2) and (\2) are easy: we take an interpolant of Y in the context of the premise. Rule (/1) must be examined in detail ((\1) is dual).

Let the rule be $TbV \to a$; $U \to c \vdash T(b/c)UV \to a$. We consider several cases.

(I) Y is contained in T or V. We take an interpolant with respect to the left premise.

(II) Y is contained in U. We take an interpolant with respect to the right premise.

(III) $Y = T_2(b/c)UV_1$, $T = T_1T_2$, $V = V_1V_2$. We take an interpolant of T_2bV_1 with respect to the left premise.

(IV) $Y = U_2V_1$, $U = U_1U_2$, $V = V_1V_2$, $U_2 \neq \Lambda$, $V_1 \neq \Lambda$. Let $U^\star$ be an interpolant of U_2 with respect to the right premise, and let $V^\star$ be an interpolant of V_1 with respect to the left premise. We take $U^\star V^\star$ as an interpolant of Y.

(V) $Y = T_2(b/c)U_1$, $T = T_1T_2$, $U = U_1U_2$, $U_2 \neq \Lambda$. Let $U^\star$ be an interpolant of U_2 with respect to the right premise, and let $T^\star$ be an interpolant of T_2b with respect to the left premise. Then, $T^\star = Sd$, $T_2 = T'T''$, and type d is an interpolant of $T''b$ with respect to the left premise. We take the string $S(d/U^\star)$ as an interpolant of Y. □

Following [24], we introduce auxiliary notions. By $\pi(a)$ we denote the set of all atomic subtypes of type a. The type a is said to be *thin*, if $\rho(p, a) = 1$, for any $p \in \pi(a)$, and the sequent $X \to a$ is said to be *thin*, if: (1) $\mathbf{L} \vdash X \to a$, (2) every type appearing in $X \to a$ is thin, (3) $\rho(p, X \to a) \in \{0, 2\}$, for any atomic type p.

Lemma 9 *Let $a_1 \dots a_n \to a_{n+1}$, $(n \geq 2)$, be a thin sequent. Then, $\pi(a_k) \subset \pi(a_{k-1}) \cup \pi(a_{k+1})$, for some $2 \leq k \leq n$.*

We skip the proof, since lemma 4 in [24] yields the same for **LP**, and **L** is a conservative subsystem of **LP**. We only note the proof uses an interpretation of **LP** in a free group. Consider the free group generated by atomic types. Define $g(a)$ by setting:

$$g(p) = p,\ g(a \star b) = g(a)g(b),\ g(a/b) = g(a)g(b)^{-1},\ g(a\backslash b) = g(a)^{-1}g(b).$$

Then, $\mathbf{LP} \vdash a_1 \dots a_n \to b$ only if $g(a_1) \dots g(a_n) = g(b)$ in the free group.

We do not know if the BR-lemma holds for **L**. We, nevertheless, prove its weaker version, restricted to sequents with atomic succedents, which suffices for the equivalence theorem. First, we prove it for thin sequents.

Lemma 10 *If $a_1 \dots a_n \to p$, $(n \geq 2)$, is a thin sequent such that $a_i \in Tp_m(P)$, for all $i = 1, \dots, n$, and $p \in P$, then there exist a number $1 \leq k < n$ and a type $b \in Tp_m(P)$ such that $\mathbf{L} \vdash a_ka_{k+1} \to b$ and*

$$\mathbf{L} \vdash a_1 \dots a_{k-1}ba_{k+2} \dots a_n \to p.$$

Proof. The proof is similar to that of lemma 6 in [24], but case (2) needs a different treatment, and an additional elimination of 'long' interpolants is involved.

If $n = 2$, then type$b = p$ fulfils the thesis. Assume $n > 2$. Let k be the number satisfying the thesis of lemma 9. Two cases are to be considered.

(1) $k < n$. Then $\pi(a_k) \subset \pi(a_{k-1}) \cup \pi(a_{k+1})$. For the set K, the cardinality of K is denoted $\#(K)$. We consider two subcases.

(1a) $\#(\pi(a_{k-1}) \cap \pi(a_k)) \geq \#(\pi(a_k) \cap \pi(a_{k+1}))$. Let Y be an interpolant of string $a_{k-1}a_k$. We show $\rho(Y) \leq \rho(a_{k-1})$. Notice that each atomic type occurs at most once in Y. We obtain:

$$\rho(Y) = \#(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + \#(\pi(a_k) - \pi(a_{k-1}) \cap \pi(a_k)) =$$

$$= \#(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + \#(\pi(a_k) \cap \pi(a_{k+1})) \leq$$

$$\leq \#(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + \#(\pi(a_{k-1}) \cap \pi(a_k)) = \#(\pi(a_{k-1})) = \rho(a_{k-1})$$

where the first equality holds by (LI3), since a_{k-1} and a_k are thin, the second equality by the inclusion from lemma 9, the inequality by the assumption of this subcase, and the remainder is obvious. Now, either Y is a single type, or $Y = ab$, where a, b are interpolants of a_{k-1}, a_k, respectively. We exclude the latter possibility. For a_{k-1} and a_k are thin, hence $\rho(a) = \rho(a_{k-1})$ and $\rho(b) = \rho(a_k)$, by (LI2), which yields $\rho(Y) > \rho(a_{k-1})$, contrary to the above. Consequently, Y is a single type which belongs to $Tp_m(P)$, again by the above. We set $b = Y$, and our thesis follows from (LI1), (LI2).

(1b) $\#(\pi(a_{k-1}) \cap \pi(a_k)) < \#(\pi(a_k) \cap \pi(a_{k+1}))$. The argument is similar; one interchanges the roles of a_{k-1} and a_{k+1}.

(2) $k = n$. Then $\pi(a_n) \subset \pi(a_{n-1}) \cup \{p\}$. Let Y be an interpolant of the string $a_{n-1}a_n$. As above, we obtain:

$$\rho(Y) = \#(\pi(a_{n-1}) - \pi(a_{n-1}) \cap \pi(a_n)) + \#(\pi(a_n) - \pi(a_{n-1}) \cap \pi(a_n)).$$

Now, $\pi(a_{n-1}) \cap \pi(a_n) \neq \emptyset$; otherwise $a_n = p$, but no sequent $Xp \to p$ with $X \neq \Lambda$ and p not appearing in X is derivable in $\mathbf{L}$. Consequently:

$$\rho(Y) \leq \rho(a_{n-1}) - \#(\pi(a_{n-1}) \cap \pi(a_n)) + 1 \leq \rho(a_{n-1}),$$

and we prove that Y is a single type fulfilling the thesis, as in case (1). □

We are ready to prove the (restricted) BR-lemma for $\mathbf{L}$.

Lemma 11 *If $\mathbf{L} \vdash X \to p$, where $l(X) \geq 2$, $X \in Tp_m(P)$, $p \in P$, then there exist types $b, c, d \in Tp_m(P)$ and strings Y, Z such that $X = YbcZ$, $\mathbf{L} \vdash bc \to d$ and $\mathbf{L} \vdash YdZ \to p$.*

Proof. Let $X \to p$ satisfy the assumptions, $X = a_1 \dots a_n$. Fix a derivation D of $X \to p$ in $\mathbf{L}$ such that all axioms (Ax) in D use atomic types only. For each atomic type q appearing in D, we form a set P_q which contains as many different copies of q, as many times axiom $q \to q$ appears in D. Next, different occurrences of this axiom are replaced by different sequents $q' \to q'$, $q' \in P_q$, which transforms D into a new derivation D'. The final sequent of D' is $a'_1 \dots a'_n \to p'$, and it is related to $X \to p$ in the same way, as D' to D. Clearly, each atomic type has precisely two, if any, occurrences in each sequent from D'. Let $b_1, \dots, b_n$ be interpolants of types $a'_1, \dots, a'_n$, respectively, in the context of the final sequent of D'. One easily sees that $b_1 \dots b_n \to p'$ is a thin sequent. By lemma 10, we find $1 \leq k < n$ and type $b' \in Tp_m(P')$ such that $\mathbf{L} \vdash b_k b_{k+1} \to b'$ and:

$$\mathbf{L} \vdash b_1 \dots b_{k-1} b' b_{k+2} \dots b_n \to p',$$

where P' denotes the join of all sets P_q, described above. Now, in the two $\mathbf{L}$-derivable sequents, mentioned in the preceding sentence, substitute q for each

$q' \in P_q$, and do it, for every atomic q appearing in D. Since $\mathbf{L}$ is closed under substitution, the two sequents give rise to new sequents $\mathbf{L} \vdash c_k c_{k+1} \to d$ and:

$$\mathbf{L} \vdash c_1 \dots c_{k-1} d c_{k+2} \dots c_n \to p,$$

where c_i is the substitution of b_i, $i = 1, \dots, n$, and d is the substitution of b'. Since b_i is an interpolant of a'_i, we have $\mathbf{L} \vdash a'_i \to b_i$, and the substitution yields $\mathbf{L} \vdash a_i \vdash c_i$, for $i = 1, \dots, n$. So, the thesis of the lemma holds by (CUT), since, clearly, $d \in Tp_m(P)$. □

Let G be an LCG. We construct a CFG Γ equivalent to G in a similar way, as at the beginning of this section. We set $V_\Gamma = V_G$, $N_\Gamma = Tp_m(P)$, where m is the maximal complexity of types appearing in I_G, and P is the set of all atomic subtypes of those types (plus s_G, if it does not appear in I_G), $s_\Gamma = s_G$, and the production rules are (G1) and (G3). Of course, **LP** may be replaced by $\mathbf{L}$, and (G2) is redundant, since no sequent $a \to p$ with $a \neq p$ is derivable in $\mathbf{L}$. $L(\Gamma) \subset L(G)$, since $\mathbf{L}$ is closed under (CUT), and the converse inclusion holds by lemma 11. So, we have just proven:

Fact 3 *If G is an LCG, then the CFG Γ constructed above is equivalent to G, that means, $L(\Gamma) = L(G)$.*

4 Main results and final comments

Let G be an LCG. The BCG G' is called *a natural expansion of* G, if $V_{G'} = V_G$, $s_{G'} = s_G$ and, for every $v \in V_G$, $I_G(v) \subset I_{G'}(v)$ and, for every $b \in I_{G'}(v)$, there exists $a \in I_G(v)$ such that $\mathbf{L} \vdash a \to b$. So, a natural expansion of the LCG G arises from G by extending the initial type-assignment: one adds new types which are $\mathbf{L}$-derivable from the types assigned to v by I_G. At the same time, one strongly impoverishes the logic: $\mathbf{L}$ is replaced by the purely applicative system $\mathbf{B}$. We shall prove that each LCG is equivalent to some natural expansion of it. Accordingly, for the purposes of sentence generation, the deductive power of the Lambek Calculus (related to lambda abstraction in semantics and to Natural Deduction in proof theory) can be restricted to the initial type assignment, while complex expressions are to be analysed by purely applicative patterns. Since only finitely many new types are affixed to the initial type assignment, then, actually, for any given LCG, only a finite fragment of $\mathbf{L}$ is significant.

Lemma 12 *If G' is a natural expansion of the LCG G, then $L(G') \subset L(G)$.*

Proof. Let $v_1 \dots v_n \in L(G')$. Then, for some $b_i \in I_{G'}(v_i)$, $i = 1, \dots, n$, we have $\mathbf{B} \vdash b_1 \dots b_n \to s_G$. By the form of $I_{G'}$, there are $a_i \in I_G(v_i)$ such that $\mathbf{L} \vdash a_i \to b_i$, $i = 1, \dots, n$. By (CUT) and the fact that $\mathbf{L}$ is stronger than $\mathbf{B}$, we get $\mathbf{L} \vdash a_1 \dots a_n \to s_G$, which yields $v_1 \dots v_n \in L(G)$. □

We are ready to prove the major result of this paper.

Theorem 1 *For every LCG G, there exists a BCG G' such that $L(G) = L(G')$ and G' is a natural expansion of G.*

PROOF. Fix an LCG G. At the end of section 3, we have constructed a CFG Γ such that $L(\Gamma) = L(G)$. Recall that $V_\Gamma = V_G$, $s_\Gamma = s_G$, nonterminal symbols of Γ are (a finite set of) types, and production rules of Γ are of the form either $d \Rightarrow bc$ with $\mathbf{L} \vdash bc \to d$, or $a \Rightarrow v$ with $a \in I_G(v)$. To each type $a \in N_\Gamma$ we assign a different atomic type p_a; we can assume $p_q = q$, for each atomic $q \in N_\Gamma$. Then, replace a with p_a in the description of Γ, for all $a \in N_\Gamma$. The resulting CFG Γ' is equivalent to Γ, hence $L(\Gamma') = L(G)$.

Now, we use results of section 2. Clearly, Γ' is in the Chomsky Normal Form. By fact 2, there exists a BCG $\mathcal{G}$ derivable from Γ' and such that $L(\mathcal{G}) = L(\Gamma')$. The BCG G' is defined as follows. $V_{G'} = V_G$, $s_{G'} = s_G$ and, for each $v \in V_G$, we set $I_{G'}(v) = I_G(v) \cup J(v)$, where $J(v)$ consists of all types b' which arise from types $b \in I_\mathcal{G}(v)$ by substituting type a for p_a at each place.

We show that G' is a natural expansion of G. Let $b' \in I_{G'}(v)$. If $b' \in I_G(v)$, then $\mathbf{L} \vdash b' \to b'$ yields the desired condition. So, assume $b' \in J(v)$. Then, b' is the substitution of a type $b \in I_\mathcal{G}(v)$. Since $\mathcal{G}$ is derivable from Γ', then there is a nonterminal symbol $p_a \in N_{\Gamma'}$ such that $\mathbf{L} \vdash p_a \to b$ and $p_a \Rightarrow v$ is a production rule of Γ'. By the construction of Γ', $a \Rightarrow v$ is a production rule of Γ, and consequently, $a \in I_G(v)$. Since $\mathbf{L}$ is closed under substitution, then $\mathbf{L} \vdash a \to b'$, which again yields the desired condition.

We show $L(G) = L(G')$. In the light of lemma 12, it suffices to prove $L(G) \subset L(G')$. Let $v_1 \dots v_n \in L(G)$. Since $L(G) = L(\mathcal{G})$, then $v_1 \dots v_n \in L(\mathcal{G})$. So, there are types $b_i \in I_\mathcal{G}(v_i)$, $i = 1, \dots, n$, such that $\mathbf{B} \vdash b_1 \dots b_n \to s_\mathcal{G}$. In this sequent, we substitute a for each atomic type p_a, which yields $\mathbf{B} \vdash b'_1 \dots b'_n \to s_G$, since $\mathbf{B}$ is closed under substitution and $s_\mathcal{G} = s_G = p_{s_G}$. Clearly, $b'_i \in J(v_i)$, for all $i = 1, \dots, n$, and consequently, $v_1 \dots v_n \in L(G')$. □

The equivalence of LCG's and CFG's (BCG's) also requires the converse statement: each CFG is equivalent to some LCG. That is an obvious consequence of the Gaifman theorem, as observed in [12, 4, 24], since each CFG is equivalent to a BCG using at most types of the form $p, p/q, (p/q)/r$, and, for sequents $X \to p$ with X consisting of types of this form, $\mathbf{B}$ is equivalent to $\mathbf{L}$. Below we give another proof, applying methods of section 2.

Fix a CFG Γ in the Chomsky Normal Form. By fact 2, there exists a BCG G derivable from Γ and such that $L(G) = L(\Gamma)$. By G' we denote the LCG which equals G except for replacing $\mathbf{B}$ with $\mathbf{L}$. We have $L(\Gamma) = L(G) \subset L(G')$, since $\mathbf{L}$ is stronger than $\mathbf{B}$. On the other hand, $L(G') \subset L(G(\Gamma)) = L(\Gamma)$, by fact 1 (the inclusion holds, since $\mathbf{L}(R^\Gamma)$ is stronger than $\mathbf{L}$ and derives all types in $I_{G'}$ from those in $I_{G(\Gamma)}$). We have shown $L(G) = L(G')$.

Similar results can be obtained for the system **L1**, by repeating the above arguments step-by-step. An alternative way is to reduce **L1**-derivability to **L**-derivability, according to the method from [9]. Namely, for each type a, one defines two finite sets of types $A(a)$ and $S(a)$ such that $\mathbf{LP1} \vdash a \to b$, for every $b \in A(a)$, and $\mathbf{LP1} \vdash b \to a$, for every $b \in S(a)$, by the following recursion:

- $A(p) = S(p) = \{p\}$, for atomic types p,
- $A(a \star b) = \{c \star d : c \in A(a), d \in A(b)\}$,

- $S(a \star b) = \{c \star d : c \in S(a), d \in S(b)\}$,
- $A(a/b) = \{c/d : c \in A(a), d \in S(b)\} \cup C(a/b)$,
- $A(a\backslash b) = \{c\backslash d : c \in S(a), d \in A(b)\} \cup C(a\backslash b)$,
- $S(a/b) = \{c/d : c \in S(a), d \in A(b)\}$,
- $S(a\backslash b) = \{c\backslash d : c \in A(a), d \in S(b)\}$,

where:

$$C(a/b) = [\ \textbf{if LP1} \vdash \Lambda \to b \ \textbf{then}\ A(a)\ \textbf{else}\ \emptyset],$$
$$C(a\backslash b) = [\ \textbf{if LP1} \vdash \Lambda \to a \ \textbf{then}\ A(b)\ \textbf{else}\ \emptyset].$$

By induction on derivations, one proves: $\mathbf{LP1} \vdash a_1 \dots a_n \to b$ if, and only if, there exist types $c_i \in A(a_i)$, $i = 1, \dots, n$, and $d \in S(b)$ such that $\mathbf{LP} \vdash c_1 \dots c_n \to d$, and the same holds with **L1** and **L**. Consequently, each categorial grammar based on **LP1** (**L**) can effectively be transformed into an equivalent grammar based on **LP** (**L**), and the latter can be transformed into an equivalent BCG with applying the methods discussed above. That yields:

Theorem 2 **LP1**-*grammars and* **L1**-*grammars are equivalent to BCG's and CFG's. Further, for each* **L1**-*grammar* G*, there exists a BCG* G' *such that* $L(G) = L(G')$ *and* G' *is a natural expansion of* G*.*

The Commutative Lambek Calculus (**CLP**) results from enriching **LP** with the permutation rule:

$$\text{(PER)}\ XabY \to c \vdash XbaY \to c,$$

and **CLP1** arises from **LP1** in the same way. Product-free fragments of these systems are denoted **CL** and **CL1**. **CL1** was used for semantic transformations of types in van Benthem [2], and **CL1** is the implication fragment of Girard's Linear Logic [17] (it coincides with the BCI-logic). It is known that the permutation closure of each context-free language can be generated by some **CL**-grammar and some **CL1**-grammar [7]. It is quite likely that the converse also holds: each **CL**-grammar generates the permutation closure of some context-free language, and similarly for **CL1**-grammars, **CLP**-grammars and **CLP1**-grammars. Unfortunately, this conjecture cannot be proven by a direct modification of the Pentus argument. The BR-lemma does not hold, in general. A counterexample is:

$$\mathbf{CLP} \vdash (p/q)/r, (r/s)/t, (t \star q)/u \to (p/s)/u,$$

which is a thin sequent, but each interpolant of two antecedent types (not necessarily adjoint) must contain four atomic subtypes, while the maximal complexity of types in this sequent is $m = 3$. One can find a product-free example, as well. We leave it as an open problem if the above conjecture holds, and if a more essential refinement of the Pentus strategy can prove it.

References

[1] Y. Bar-Hillel, C. Gaifman and E. Shamir, On categorial and phrase structure grammars, *Bull. Res. Council Israel* F 9, (1960), 155-166.

[2] J. van Benthem, *Language in Action. Categories, Lambdas and Dynamic Logic*, North-Holland, Amsterdam, 1991.

[3] W. Buszkowski, Some decision problems in the theory of syntactic categories, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 28, (1982), 539-548.

[4] W. Buszkowski, The Equivalence of Unidirectional Lambek Categorial Grammars and Context-Free Grammars, *ibidem* 31, (1985), 369-384.

[5] W. Buszkowski, Completeness Results for Lambek Syntactic Calculus, *ibidem* 32, (1986), 13-28.

[6] W. Buszkowski, Generative Capacity of Nonassociative Lambek Calculus, *Bull. Polish Academy of Sciences. Mathematics* 34, (1986), 507-516.

[7] W. Buszkowski, Generative Power of Categorial Grammars, in [23].

[8] W. Buszkowski, Gaifman's Theorem on Categorial Grammars revisited, *Studia Logica* 47, (1988), 23-33.

[9] W. Buszkowski, On Generative Capacity of the Lambek Calculus, in [15].

[10] W. Buszkowski, W. Marciszewski and J. van Benthem (eds.), *Categorial Grammar*, J. Benjamins, Amsterdam, 1988.

[11] N. Chomsky, Formal properties of grammars, in: R. D. Luce et al. (eds.), *Handbook of Mathematical Psychology*, vol. 2, Wiley, New York, 1973.

[12] J. M. Cohen, The equivalence of two concepts of categorial grammar, *Information and Control* 10, (1967), 475-484.

[13] K. Došen and P. Schroeder-Heister (eds.), *Substructural Logics*, Oxford University Press, Oxford, 1993.

[14] J. M. Dunn, Partial Gaggles Applied to Logics with Restricted Structural Rules, in [13].

[15] J. van Eijck (ed.), *Logics in AI*, Lecture Notes in Artificial Intelligence, Springer, Berlin, 1991.

[16] D. Gabbay, *Labelled Deductive Systems I*, CIS-München, Munich, 1991.

[17] J. Y. Girard, Linear Logic, *Theoretical Computer Science* 50, (1987), 1-102.

[18] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading Mass., 1979.

[19] M. Kandulski, The Equivalence of Nonassociative Lambek Categorial Grammars and Context-Free Grammars, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 34, (1988), 41-52.

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

[21] J. Lambek and P. J. Scott, *Introduction to higher-order categorical logic*, Cambridge University Press, Cambridge, 1986.

[22] M. Moortgat, *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht, 1988.

[23] R. T. Oehrle, E. Bach and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, D. Reidel, Dordrecht, 1988.

[24] M. Pentus, Lambek Grammars are Context-Free, *Prepublication Series: Mathematical Logic and Theoretical Computer Science* 8, Steklov Mathematical Institute of The Russian Academy of Sciences, Moscow, 1992.

[25] V. Pratt, Action Logic and pure induction, in [15].

[26] D. Roorda, *Resource Logics: Proof-theoretical Investigations*, Ph.D. Thesis, Faculty of Mathematics and Computer Science, University of Amsterdam, 1991.