Institute for Logic, Language and Computation

ON THE EQUIVALENCE OF LAMBEK CATEGORIAL GRAMMARS AND BASIC CATEGORIAL GRAMMARS

Wojciech Buszkowski

ILLC Prepublication Series
for Logic, Semantics and Philosophy of Language LP-93-07

University of Amsterdam

The ILLC Prepublication Series

1990 *Logic, Semantics and Philosophy of Language*

LP-90-01 Jaap van der Does — A Generalized Quantifier Logic for Naked Infinitives
LP-90-02 Jeroen Groenendijk, Martin Stokhof — Dynamic Montague Grammar
LP-90-03 Renate Bartsch — Concept Formation and Concept Composition
LP-90-04 Aarne Ranta — Intuitionistic Categorial Grammar
LP-90-05 Patrick Blackburn — Nominal Tense Logic
LP-90-06 Gennaro Chierchia — The Variablity of Impersonal Subjects
LP-90-07 Gennaro Chierchia — Anaphora and Dynamic Logic
LP-90-08 Herman Hendriks — Flexible Montague Grammar
LP-90-09 Paul Dekker — The Scope of Negation in Discourse, towards a Flexible Dynamic Montague grammar
LP-90-10 Theo M.V. Janssen — Models for Discourse Markers
LP-90-11 Johan van Benthem — General Dynamics
LP-90-12 Serge Lapierre — A Functional Partial Semantics for Intensional Logic
LP-90-13 Zhisheng Huang — Logics for Belief Dependence
LP-90-14 Jeroen Groenendijk, Martin Stokhof — Two Theories of Dynamic Semantics
LP-90-15 Maarten de Rijke — The Modal Logic of Inequality
LP-90-16 Zhisheng Huang, Karen Kwast — Awareness, Negation and Logical Omniscience
LP-90-17 Paul Dekker — Existential Disclosure, Implicit Arguments in Dynamic Semantics

Mathematical Logic and Foundations

ML-90-01 Harold Schellinx — Isomorphisms and Non-Isomorphisms of Graph Models
ML-90-02 Jaap van Oosten — A Semantical Proof of De Jongh's Theorem
ML-90-03 Yde Venema — Relational Games
ML-90-04 Maarten de Rijke — Unary Interpretability Logic
ML-90-05 Domenico Zambella — Sequences with Simple Initial Segments
ML-90-06 Jaap van Oosten — Extension of Lifschitz' Realizability to Higher Order Arithmetic, and a Solution to a Problem of F. Richman
ML-90-07 Maarten de Rijke — A Note on the Interpretability Logic of Finitely Axiomatized Theories
ML-90-08 Harold Schellinx — Some Syntactical Observations on Linear Logic
ML-90-09 Dick de Jongh, Duccio Pianigiani — Solution of a Problem of David Guaspari
ML-90-10 Michiel van Lambalgen — Randomness in Set Theory
ML-90-11 Paul C. Gilmore — The Consistency of an Extended NaDSet

Computation and Complexity Theory

CT-90-01 John Tromp, Peter van Emde Boas — Associative Storage Modification Machines
CT-90-02 Sieger van Denneheuvel, Gerard R. Renardel de Lavalette — A Normal Form for PCSJ Expressions
CT-90-03 Ricard Gavaldà, Leen Torenvliet, Osamu Watanabe, José L. Balcázar — Generalized Kolmogorov Complexity in Relativized Separations
CT-90-04 Harry Buhrman, Edith Spaan, Leen Torenvliet — Bounded Reductions
CT-90-05 Sieger van Denneheuvel, Karen Kwast — Efficient Normalization of Database and Constraint Expressions
CT-90-06 Michiel Smid, Peter van Emde Boas — Dynamic Data Structures on Multiple Storage Media, a Tutorial
CT-90-07 Kees Doets — Greatest Fixed Points of Logic Programs
CT-90-08 Fred de Geus, Ernest Rotterdam, Sieger van Denneheuvel, Peter van Emde Boas — Physiological Modelling using RL
CT-90-09 Roel de Vrijer — Unique Normal Forms for Combinatory Logic with Parallel Conditional, a case study in conditional rewriting

Other Prepublications

X-90-01 A.S. Troelstra — Remarks on Intuitionism and the Philosophy of Mathematics, Revised Version
X-90-02 Maarten de Rijke — Some Chapters on Interpretability Logic
X-90-03 L.D. Beklemishev — On the Complexity of Arithmetical Interpretations of Modal Formulae
X-90-04 — Annual Report 1989
X-90-05 Valentin Shehtman — Derived Sets in Euclidean Spaces and Modal Logic
X-90-06 Valentin Goranko, Solomon Passy — Using the Universal Modality: Gains and Questions
X-90-07 V.Yu. Shavrukov — The Lindenbaum Fixed Point Algebra is Undecidable
X-90-08 L.D. Beklemishev — Provability Logics for Natural Turing Progressions of Arithmetical Theories
X-90-09 V.Yu. Shavrukov — On Rosser's Provability Predicate
X-90-10 Sieger van Denneheuvel, Peter van Emde Boas — An Overview of the Rule Language RL/1
X-90-11 Alessandra Carbone — Provable Fixed points in $I\Delta_0+\Omega_1$, revised version
X-90-12 Maarten de Rijke — Bi-Unary Interpretability Logic
X-90-13 K.N. Ignatiev — Dzhaparidze's Polymodal Logic: Arithmetical Completeness, Fixed Point Property, Craig's Property
X-90-14 L.A. Chagrova — Undecidable Problems in Correspondence Theory
X-90-15 A.S. Troelstra — Lectures on Linear Logic

1991 *Logic, Semantics and Philosophy of Langauge*

LP-91-01 Wiebe van der Hoek, Maarten de Rijke — Generalized Quantifiers and Modal Logic
LP-91-02 Frank Veltman — Defaults in Update Semantics
LP-91-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions
LP-91-04 Makoto Kanazawa — The Lambek Calculus enriched with Additional Connectives
LP-91-05 Zhisheng Huang, Peter van Emde Boas — The Schoenmakers Paradox: Its Solution in a Belief Dependence Framework
LP-91-06 Zhisheng Huang, Peter van Emde Boas — Belief Dependence, Revision and Persistence
LP-91-07 Henk Verkuyl, Jaap van der Does — The Semantics of Plural Noun Phrases
LP-91-08 Víctor Sánchez Valencia — Categorial Grammar and Natural Reasoning
LP-91-09 Arthur Nieuwendijk — Semantics and Comparative Logic
LP-91-10 Johan van Benthem — Logic and the Flow of Information

Mathematical Logic and Foundations

ML-91-01 Yde Venema — Cylindric Modal Logic
ML-91-02 Alessandro Berarducci, Rineke Verbrugge — On the Metamathematics of Weak Theories
ML-91-03 Domenico Zambella — On the Proofs of Arithmetical Completeness for Interpretability Logic
ML-91-04 Raymond Hoofman, Harold Schellinx — Collapsing Graph Models by Preorders
ML-91-05 A.S. Troelstra — History of Constructivism in the Twentieth Century
ML-91-06 Inge Bethke — Finite Type Structures within Combinatory Algebras
ML-91-07 Yde Venema — Modal Derivation Rules
ML-91-08 Inge Bethke — Going Stable in Graph Models
ML-91-09 V.Yu. Shavrukov — A Note on the Diagonalizable Algebras of PA and ZF
ML-91-10 Maarten de Rijke, Yde Venema — Sahlqvist's Theorem for Boolean Algebras with Operators
ML-91-11 Rineke Verbrugge — Feasible Interpretability
ML-91-12 Johan van Benthem — Modal Frame Classes, revisited

Computation and Complexity Theory

CT-91-01 Ming Li, Paul M.B. Vitányi — Kolmogorov Complexity Arguments in Combinatorics
CT-91-02 Ming Li, John Tromp, Paul M.B. Vitányi — How to Share Concurrent Wait-Free Variables
CT-91-03 Ming Li, Paul M.B. Vitányi — Average Case Complexity under the Universal Distribution Equals Worst Case Complexity
CT-91-04 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence
CT-91-05 Sieger van Denneheuvel, Karen Kwast — Weak Equivalence for Constraint Sets
CT-91-06 Edith Spaan — Census Techniques on Relativized Space Classes
CT-91-07 Karen L. Kwast — The Incomplete Database
CT-91-08 Kees Doets — Levationis Laus
CT-91-09 Ming Li, Paul M.B. Vitányi — Combinatorial Properties of Finite Sequences with high Kolmogorov Complexity
CT-91-10 John Tromp, Paul Vitányi — A Randomized Algorithm for Two-Process Wait-Free Test-and-Set
CT-91-11 Lane A. Hemachandra, Edith Spaan — Quasi-Injective Reductions
CT-91-12 Krzysztof R. Apt, Dino Pedreschi — Reasoning about Termination of Prolog Programs

Computational Linguistics

CL-91-01 J.C. Scholtes — Kohonen Feature Maps in Natural Language Processing
CL-91-02 J.C. Scholtes — Neural Nets and their Relevance for Information Retrieval

Institute for Logic, Language and Computation
Plantage Muidergracht 24
1018TV Amsterdam
Telephone 020-525.6051, Fax: 020-525.5101

ON THE EQUIVALENCE OF LAMBEK CATEGORIAL GRAMMARS AND BASIC CATEGORIAL GRAMMARS

Wojciech Buszkowski
Department of Mathematics
Adam Mickiewicz University, Poznań

ILLC Prepublications
for Logic, Semantics and Philosophy of Language
ISSN 0928-3307

Coordinating editor: Dick de Jongh

received July 1993

On the Equivalence of Lambek Categorial Grammars and Basic Categorial Grammars[1]

Wojciech Buszkowski
Institute of Mathematics
Adam Mickiewicz University
Matejki 48/49
Poznań
60-769 Poland
email: buszko@plpuam11.bitnet

Abstract

Due to M. Pentus [19], we know that LCG's are weakly equivalent to Context-Free Grammars, hence also to BCG's (by Gaifman's theorem [2]). Here we show that, for any product-free LCG G, there is an equivalent BCG G' which results from expanding the initial type assignment of G by means of some Lambek derivable formulae, as it has been expected by many authors and erroneously proven by some of them. Our construction uses a modification of Pentus' argument and the interpretation of Gaifman's theorem on the basis of the Lambek Calculus given in [10]. The latter is presented here more carefully than in [10]; as a result, a new proof of Gaifman's theorem is obtained in which the Lambek Calculus is essentially involved.

1. Introduction and preliminaries

Categorial grammars are formal grammars which describe a language by assigning logical types to atoms and deriving types of complexes from types of atoms by means of some systems of type change. These systems produce sequents $a_1 \dots a_n \rightarrow a$, where $a_1,\dots,a_n$ and a are types. Formally, 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 distinguished atomic type, and R_G is a system of type change; one refers to these four components as *the lexicon, the initial type assignment, the principal type,* and *the system,* respectively, of the grammar G. We say that G assigns type a to string $v_1 \dots v_n$ ($v_i \in V_G$), if, for some $a_i \in I_G(v_i)$, $i=1,\dots,n$, sequent $a_1 \dots a_n \rightarrow a$ is derivable in R_G. The set $L(G)$, called *the language of* G, consists of all the strings on V_G which are assigned

[1]The paper has been written during the author's stay at the University of Amsterdam in summer 1993 due to a grant from the Commission of the European Communities for the project "Mathematical Linguistics and Proof Theory" (no. 1506).

type s_G by G. Two grammars are said to be *equivalent*, if they yield the same language (this notion is also applied to other kinds of formal grammars, provided the concept of language is defined for them).

Types are formed out of some constants (*atomic types*) by means of binary symbols /, \, and *, called *right residuation, left 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 (also with subscripts, primes, etc.). *Basic Categorial Grammars* (BCG's) admit the system **B** which deals with product-free types and can be axiomatized as follows:

(Ax) $a \rightarrow a,$
(/1) $XaZ \rightarrow c, Y \rightarrow b \vdash X(a/b)YZ \rightarrow c,$
(\1) $XbZ \rightarrow c, Y \rightarrow a \vdash XY(a\backslash b)Z \rightarrow c.$

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

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

and consequently, BCG's are precisely the categorial grammars in the sense of Bar-Hillel et al. [2].

Lambek Categorial Grammars (LCG's) are based on the system **L** which results from enriching **B** with two additional rules:

(/2) $Xb \rightarrow a \vdash X \rightarrow a/b,$
(\2) $aX \rightarrow b \vdash X \rightarrow a\backslash b,$

where X is nonempty (dropping this constraint leads to a stronger system **L1**). The original *Lambek Calculus* [16] admits types with product and can be axiomatized by (Ax), (/1),(\1),(/2),(\2) together with the product-introduction rules:

(*1) $XabY \rightarrow c \vdash X(a{*}b)Y \rightarrow c,$
(*2) $X \rightarrow a, Y \rightarrow b \vdash XY \rightarrow (a{*}b);$

we denote the latter system by **LP**, and **LP1** is defined in a similar way as **L1**. In each of the four variants of the Lambek Calculus, axioms (Ax) can be restricted to atomic types a. Their axiomatizations are Gentzen-style ones without structural rules, and **B** admits introduction-in-antecedent only. In particular, each of the systems mentioned above is decidable and closed under *the cut rule* :

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

which has been established for **LP** by Lambek [16] (the decidability of **B** may be credited as far as to Ajdukiewicz (1935)).

The Lambek Calculus and its sub- and super-systems are closely related to several issues of current interest in logic, as e.g. linear logics (Girard), concatenation logics (Gabbay), action logics (Pratt), substructural logics

(Došen), and some more philosophical topics like natural logic and inference based on monotonicity (see Sanchéz [21] for a historico-logical analysis). In this paper we follow the linguistic thread applying them to the account of type change in syntax and semantics of natural language (E. Bach, J. van Benthem, B. Hall-Partee, M. Moortgat, R. T. Oehrle, F. Zwarts), which justifies their suitability for categorial grammars (this linguistic perspective was in focus in the research of our Poznan group in that area, since the late seventies, though our mathematical investigations were concerned with logical fundamentals as well). A profound discussion of logical aspects of these systems is given in van Benthem [3], [5], while the linguistic side is extensively studied in e.g. Moortgat [17] and Oehrle et al. [18].

From the standpoint of type-theoretic semantics, **B** is a purely applicative system (rules (R/) and (R\) correspond to function application), while Lambek-style systems also employ some forms of lambda abstraction (see [5]). A problem which has quite early appeared in the history of the discipline is whether introducing lambda abstraction essentially affects generative capacity. In other words, the question is of whether LCG's are equivalent to or stronger than BCG's with respect to the generation of string languages. In [2], BCG's are shown to be equivalent to Context-Free Grammars (CFG's) (we refer to that result as *the Gaifman theorem*), and the authors conjecture the same equivalence holds for LCG's. This conjecture, repeated in Chomsky [12], is now addressed to as *the Chomsky conjecture* by some authors. Since the late sixties, there were undertaken several attempts to prove the conjecture. Cohen [13] shows that each BCG is equivalent to some LCG, and presents a proof of the converse statement which contains essential errors (see [7]). There were obtained partial results in this direction: for LCG's restricted to /-types or \-types [7] and for LCG's of order at most 2 [11]. There was also established the equivalence (even a kind of strong equivalence) of BCG's and categorial grammars based on the Nonassociative Lambek Calculus [8], [15]. Finally, Pentus [19] has found a splendid proof for the full calculus **LP** (also for **LP1**), using quite fundamental logical properties of these systems and a bit of combinatorics. It follows from Pentus' theorem that each LCG is equivalent to some CFG, hence to some BCG, and the same holds for categorial grammars based on **L1**, **LP** and **LP1**. (No kind of strong equivalence is possible here, by structural completeness of Lambek-style systems; see [9].)

The way Pentus proceeds in his proof is completely different from that advocated by Cohen and successfully used in [8], [15] for the nonassociative case. Let us refer to the latter as *the natural way*. It consists in the following. Given an LCG G, one looks for a BCG G' such that $L(G) = L(G')$. First, one constructs an infinite grammar G^* whose lexicon and principal type are those of G, while its system is **B**, and its initial type assignment I^* is an infinite extension of I_G (consequently, G^* is "an infinite BCG"). Namely, for any $v \in V_G$, $I^*(v)$ consists of all types b such that $a \rightarrow b$ is derivable in **L**, for some $a \in I_G(v)$. It is easy to see $L(G) = L(G^*)$. For, $\supseteq$ holds, since **B** is a subsystem of **L**, and **L** is closed under (CUT). To show $\subseteq$ it suffices to observe that $\mathbf{L} \vdash aX \rightarrow b$ entails $\mathbf{L} \vdash a \rightarrow b/X$, and $\mathbf{B} \vdash (b/X)X \rightarrow b$, where b/X is recursively defined, as follows:

(1) $b/\Lambda = b$, $b/(Xc) = (b/c)/X$ (Λ denotes the empty string).

Now, one wants to transform G^* into a real (i.e. finite) BCG by restricting I^* to its finite part without changing generative capacity. So, a *natural* BCG G' equivalent to the given LCG G must satisfy the following conditions: $V_{G'} = V_G, I_{G'}(v) \subseteq I^*(v)$, for all $v \in V_G$, $s_{G'} = s_G$, and, of course, $L(G') = L(G)$.

Cohen [13] constructs $I_{G'}(v)$ by affixing to $I_G(v)$ all the types which result from applying precisely once an axiomatic rule of his axiomatization of **L** (restricted to formulae $a \to b$) to each type from $I_G(v)$. As shown in [7], this yields, in general, a too poor initial type assignment, and one merely gets $L(G') \subset L(G)$. Recently, E. König attempted the same route with a much richer stock of rules, but her arguments look sticky in many details. The present paper provides a rather smooth proof of the existence of a natural BCG equivalent to any given LCG. We essentially use Pentus' construction of a CFG equivalent to a given LCG and a construction of a BCG equivalent to a given CFG. The latter construction is based on the Lambek Calculus, while all the earlier approaches to the Gaifman theorem (and the closely related Greibach normal form theorem from the theory of Context-Free Grammars) use some purely combinatorial tools. Actually, for our main construction of a natural BCG equivalent to a given LCG, it is quite crucial to derive in the Lambek Calculus what is needed for the Gaifman theorem.

The paper consists of four sections. In section 2 the Gaifman theorem is proven with the aid of L; actually, the proof differs from the original proof from [2] but resembles the simplified proof given by Gladkij [14]. As we have already noticed in [9], [10], Gladkij's key construction, though typically combinatorial in spirit, admits nonetheless an interesting interpretation on the basis of L, and we draw here further consequences of this fact. Section 3 adapts the Pentus theorem to the product-free case; in particular, Roorda's Interpolation Lemma [20] is adapted to L (M. Pentus announced independent results in this direction). The main construction is provided in section 4 which also contains some final comments.

2. The Gaifman theorem

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*, each of them is of one of the forms:

(2) $p \Rightarrow p_1 \ldots p_n$, where $p, p_1, \ldots, p_n \in N_\Gamma$,
(3) $p \Rightarrow v$, where $p \in N_\Gamma$, $v \in V_\Gamma$.

We symbolize nonterminal symbols of a CFG and atomic types of a categorial grammar by the same letters, since we identify them in what follows. The relation $p \Rightarrow_\Gamma X$, where $p \in N_\Gamma$, $X \in N_\Gamma^+$, is recursively defined as follows:

(4) $p \Rightarrow_\Gamma p$, for all $p \in N_\Gamma$,
(5) if $p_i \Rightarrow_\Gamma X_i$, for $i = 1, \ldots, n$, then $p \Rightarrow_\Gamma X_1 \ldots X_n$,

for any production rule (2) from R_Γ. *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 $p_1, \dots p_n$, there holds $s_\Gamma \Rightarrow_\Gamma p_1 \dots p_n$, and $p_i \Rightarrow v_i$ is in R_Γ, for $i = 1, \dots, n$. It is well known that each CFG is equivalent to a CFG in the Chomsky Normal Form whose production rules (2) have always $n = 2$.

The Gaifman theorem 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 easy to prove. Given a BCG G, we obtain an equivalent CFG Γ in the following way. The terminal symbols of Γ are the symbols from V_G. The nonterminal symbols of Γ are all subtypes of the types appearing in I_G. The initial symbol of Γ equals the principal type of G. The production rules (2) of Γ are simply the rules (R/), (R\) restricted to nonterminal symbols of Γ and written in the reverse direction. The production rules (3) of Γ are all clauses $a \Rightarrow v$ such that $v \in V_G$ and $a \in I_G(v)$.

Statement (II) is much less trivial; it is equivalent to the Greibach normal form theorem in the theory of CFG's (so, the Greibach theorem is due to H. Gaifman). In this section we prove it with the aid of the Lambek Calculus (see also [10] for an algebraic proof based on congruences and transformations in the algebra of phrase structures).

We need, actually, not the pure Lambek Calculus but its axiomatic extensions, first introduced in [6]. Let R be a set of product-free formulae $X \rightarrow a$ ($X \neq \Lambda$). By L(R) we denote the system axiomatized by axioms (Ax) and all the formulae from R (as new axioms) and the inference rules of L with (CUT). An equivalent Gentzen-style axiomatization can be given as follows. First, observe that each formula is equivalent to a formula $X \rightarrow p$ (p is atomic!) on the basis of L (equivalence means mutual derivability). So, we assume R consists of formulae of the latter form. The system GL(R) is axiomatized by (Ax), (/1), (\1), (/2), (\2), and the special rules:

(R.Ax) $X_1 \rightarrow a_1, \dots X_n \rightarrow a_n \vdash X_1 \dots X_n \rightarrow p$,

one for each formula $a_1 \dots a_n \rightarrow p$ from R.

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

Proof. The proof goes 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 (R.Ax) cannot be the second premise of (CUT), if a in the first premise results from (/1) or (\1).

Corollary 1. The same formulae are derivable in GL(R) and L(R).

Proof. By (CUT), L(R) is closed under each rule (R.Ax), hence it is not weaker than GL(R). By (Ax) and (R.Ax), each formula from R is derivable in GL(R), hence we obtain the converse, using lemma 1.

Let us note that corollary 1 does not imply the decidability of systems L(R), even for finite R (rules (R.Ax) may forget information). It has been shown in [6] that every recursively enumerable language can be generated by a categorial grammar based on some system L(R) with R finite.

We are interested here in especially simple sets R which consist of finitely many formulae of the form:

(7)$p_1 \ldots p_n \rightarrow p$,

which are directly related to production rules (2). For those sets R, systems L(R) are decidable (see [6]). By R^Γ we denote the set of all formulae (7) corresponding to production rules (2) of the CFG Γ.

Lemma 2. For any $p, p_1, \ldots, p_n \in N_\Gamma$, $p \Rightarrow_\Gamma p_1 \ldots p_n$ if, and only if, L(R^Γ)$\vdash p_1 \ldots p_n \rightarrow p$.

Proof. Since L(R^Γ) admits (CUT), then "only if" holds. For "if", it is enough to notice that each derivation of $p_1 \ldots p_n \rightarrow p$ in GL(R^Γ) uses at most (Ax) and (R.Ax), hence it is simply 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 L(R^Γ) 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 (atomic types) p such that (3) belongs to R_Γ . As an immediate consequence of lemma 2, we obtain:

Corollary 2. $L(G(\Gamma)) = L(\Gamma)$.

Let G be a BCG. We say that G *is derivable from* Γ if the lexicon and the principal type of G are those of $G(\Gamma)$, and the initial type assignment of G fulfils the condition:

(8) if $a \in I_G(v)$, then L(R^Γ)$\vdash p \rightarrow a$, for some $p \in I_{G(\Gamma)}(v)$,

for any $v \in V_G$. If G is derivable from Γ then $L(G) \subseteq L(G(\Gamma))$ (since L(R^Γ) admits (CUT) and is stronger than B), hence, by corollary 2, we obtain:

Lemma 3. If a BCG G is derivable from a CFG Γ, then $L(G) \subseteq L(\Gamma)$.

Accordingly, we shall succeed in constructing a BCG equivalent to a given CFG Γ, if we find a BCG G derivable from Γ such that $L(\Gamma) \subseteq L(G)$. To accomplish this goal we need the following properties of the Lambek Calculus:

(9) if $L(R) \vdash qr \to p$, then $L(R) \vdash r \to q \backslash p$,
(10) $L \vdash q \backslash p \to (q \backslash t)/(p \backslash t)$,
(11) $L \vdash q \to p/(q \backslash p)$,
(12) if $L(R) \vdash a \to b$, then $L(R) \vdash a/c \to b/c$,

for all types p, q, r, t, a, b, c (p, q, r, t need not be atomic). (9) holds by (\2). (10) follows from $L \vdash q(q \backslash p)(p \backslash t) \to t$, by (\2) and (/2). (11) is a consequence of $L \vdash p(p \backslash q) \to q$, by (/2). For (12) $(a/c)c \to a$ is derivable in L, hence $a \to b$ entails $(a/c)c \to b$, by (CUT), which yields $a/c \to b/c$, by (/2).

Now, fix a CFG Γ in the Chomsky Normal Form. First, we define a mapping I which to any nonterminal symbol of Γ assigns a finite set of types and satisfies the condition:

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

We set $I(p) = I_1(p) \cup I_2(p)$, where I_1 and I_2 are defined, as follows. For any production rule:

(14) $p \Rightarrow qr$,

from R_Γ , we put types:

(15) $q \backslash p$ and $(q \backslash t)/(p \backslash t)$, for all $t \in N_\Gamma$,

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 the type:

(16) $a/(q \backslash p)$,

into $I_2(q)$. This finishes the construction of I. Observe that (13) holds, by (9)-(12). We only consider type (16). Since $L(R^{\Gamma}) \vdash p \to a$, as $a \in I_1(p)$, then $L(R^{\Gamma}) \vdash p/(q \backslash p) \to a/(q \backslash p)$, by (12), hence $L(R^{\Gamma}) \vdash q \to a/(q \backslash p)$, by (11) and (CUT). Second, we define a BCG G derivable from Γ by setting: $I_G(v)$, for $v \in V_G$, equals the set of all types a such that, for some $p \in I_{G(\Gamma)}(v)$ (that means, $p \Rightarrow v$ is in R_Γ), $a \in I(p)$.

We must show $L(\Gamma) \subseteq L(G)$. To do it we need some simple properties of Γ-derivability. Observe that condition (5) takes the form:

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

for any production rule (14) from R_Γ. A derivation in Γ is said to be *regular*, if $Y = r$ in each application of rules (17) (that means, only the left-hand part of the derivation tree is to be expanded). The next lemma exhibits regular subderivations of each Γ-derivation.

Lemma 4. 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 $p \Rightarrow_\Gamma q q_1 \ldots q_k$ 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, for some production rule $p\Rightarrow rs$, there hold $r\Rightarrow_\Gamma qY$ and $s\Rightarrow_\Gamma Z$, for some strings Y, Z such that $X=YZ$. Since $Z\neq\Lambda$, then Y is shorter than X. By induction, there are $k\geq 0$, $q_1,\dots,q_k$ and $X_1,\dots,X_k$ such that $Y=X_1\dots X_k$, $q_i\Rightarrow_\Gamma X_i$, for $i=1,\dots,k$, and $r\Rightarrow_\Gamma qq_1\dots q_k$ has a regular derivation. We take $q_{k+1}=s$ and $X_{k+1}=Z$.

Lemma 5. 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 $\mathbf{B}\vdash bb_1\dots b_k\rightarrow a$, and rule (\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 a sequence of production rules:

(18) $p\Rightarrow r_kq_k$, $r_k\Rightarrow r_{k-1}q_{k-1}$, ... , $r_3\Rightarrow r_2q_2$, $r_2\Rightarrow qq_1$,

for some nonterminal symbols $r_2,\dots,r_k$. By (15), we obtain the following assignment:

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

denote these types by $b_k,\dots,b_1$, respectively. Evidently:

(20) $\mathbf{B}\vdash b_1\dots b_k\rightarrow q\backslash p$,

and (R\) is not applied in this derivation. Now, choose $a\in I_1(p)$. By (16), we have $a/(q\backslash p)\in I_2(q)$, which yields the thesis with $b=a/(q\backslash p)$. Case $k=1$ is particular: $p\Rightarrow qq_1$ is the only rule in (18), and we set $b_1=q\backslash p$ and b as above.

Thus, the BCG G constructed above can simulate regular derivations in Γ. We show it is so for arbitrary derivations.

Lemma 6. 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 $\mathbf{B}\vdash c_1\dots c_n\rightarrow a$, and rule (\1) is not applied in the latter derivation.

Proof. Induction on n. For $n=1$, the derivation is regular, hence lemma 5 yields the thesis. Assume $n>1$. By lemma 4, there are $k, q_1,\dots,q_k$ and $X_1,\dots,X_k$ such that $p_2\dots p_n=X_1\dots X_k$ (so, $k\neq 0$), $q_i\Rightarrow_\Gamma X_i$, for $i=1,\dots,k$, and $p\Rightarrow_\Gamma p_1q_1\dots q_k$ has a regular derivation. Choose $a\in I_1(p)$. By lemma 5, there are types $c_1\in I(p_1)$ and $b_i\in I_1(q_i)$, $i=1,\dots,k$, such that $\mathbf{B}\vdash c_1b_1\dots b_k\rightarrow a$ without (\1). By induction, since $b_i\in I_1(q_i)$ and $q_i\Rightarrow_\Gamma X_i$, then we can find a string Y_i, of types assigned by I to the corresponding symbols from X_i, such that $\mathbf{B}\vdash Y_i\rightarrow b_i$ without (\1). Consequently, $\mathbf{B}\vdash c_1Y_1\dots Y_k\rightarrow a$ holds by (CUT), and we set $Y_1\dots Y_k=c_2\dots c_n$ (clearly, (\1) is not applied).

Corollary 3. $L(\Gamma)\subseteq L(G)$.

Using lemma 3, we infer $L(G) = L(\Gamma)$, so G is a BCG equivalent to Γ. As a matter of fact, we have proven statement (II) in its full strength. For rule (\1) need not be used in G to establish corollary 3. Accordingly, we may assume the system of G skips this rule (then, of course, lemma 3 holds as well). In **B** lacking (\1) types of the form $a\backslash b$ are treated as atomic types (they never appear as functors in reductions (R\)), hence they can be replaced by different atomic types. Write $p^{\,q}$ for $p\backslash q$ in the definition of I, and regard $p^{\,q}$ as an atomic type. Thus, (15) and (16) obtain the form:

(15') $p^{\,q}, p^{\,t}/q^{\,t}$, (16') $a/q^{\,p}$, where $a = s_\Gamma$ or a is of the form (15'),

which fulfils the constraint in (II). The latter move, however, makes the connection with the Lambek Calculus less transparent, while to illuminate this connection has been our main goal in this section. The above proof of the Gaifman theorem has been presented in detail just to illustrate the possibility the Lambek Calculus can serve as a device which transforms one grammar (here a CFG) into another grammar (here a BCG) of the same language. We believe linguistics will exploite this function of Lambek style systems in further developments, since perspectives seem much promising.

3. Interpolation and binary reductions in L

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

(21) $\rho(a_1 \ldots a_n) = \rho(a_1) + \ldots + \rho(a_n)$, $\rho(X \rightarrow a) = \rho(X) + \rho(a)$.

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 formed out of atomic types from P and such that $\rho(a) \leq n$, and $Tp_n(P)$ denotes the restriction of the latter to product-free types.

The key lemma in Pentus [19] is the following: *For any set P and any number $n \geq 1$, if* **LP** $\vdash X \rightarrow a$, where $l(X) \geq 2$, $X \in TP_n(P)^+$, $a \in TP_n(P)$, *then there exist types $b,c,d \in TP_n(P)$ and strings Y, Z such that $X = YbcZ$,* **LP** $\vdash bc \rightarrow d$ and **LP** $\vdash YdZ \rightarrow a$. We refer to this lemma as the binary reduction lemma (*the BR-lemma*). The BR-lemma had been proven in [7], [11] for some special families of product-free types only, while M. Pentus succeeded in establishing it for arbitrary types (with product), by a deeper penetration into the logical structure of **LP**.

The BR-lemma yields one direction of the equivalence between categorial grammars based on **LP** (**LP**-grammars) and CFG's, namely, the fact that each **LP**-grammar is equivalent to some CFG. Let G be an **LP**-grammar. We construct a CFG Γ in the following way. Let P be the set of all atomic subtypes of the types appearing in I_G (also $s_G \in P$), and let n be the maximal complexity of the latter types. We set $V_\Gamma = V_G$, $N_\Gamma = TP_n(P)$ and $s_\Gamma = s_G$. The production rules of Γ are:

(22) $d \Rightarrow bc$, for $b,c,d \in N_\Gamma$ such that $\mathbf{LP} \vdash bc \to d$,
$a \Rightarrow b$, for $a,b \in N_\Gamma$ such that $\mathbf{LP} \vdash b \to a$,
(23) $a \Rightarrow v$, for $a \in N_\Gamma$, $v \in V_\Gamma$ such that $a \in I_G(v)$.

$L(\Gamma) \subseteq L(G)$ holds, since **LP** is closed under (CUT). $L(G) \subseteq L(\Gamma)$ holds by the BR-lemma (the second rules in (22) are used for one-step derivations only).

The other direction of the equivalence in question is an easy consequence of the Gaifman theorem. Let Γ be a CFG. By statement (II) from section 2, Γ is equivalent to a BCG G whose initial type assignment uses at most types of the form p, p/q, $(p/q)/r$. These types are of order not greater than 1, where the order of product-free types is defined, as follows:

(24) $ord(p) = 0$, $ord(a/b) = ord(b \backslash a) = \max(ord(a), ord(b) + 1)$.

Now, for any sequent $X \to p$ such that all types in X are of order not greater than 1, there holds:

(25) if $\mathbf{LP} \vdash X \to p$, then $\mathbf{B} \vdash X \to p$,

since rules (/2), (\2), (*1) and (*2) are not used in any derivation of $X \to p$ in **LP**. Consequently, **LP** is equivalent to **B** for such sequents, and the **LP**-grammar G' which differs from G in just admitting **LP** instead of **B** as its system is equivalent to G. So, G' must also be equivalent to Γ.

An alternative argument for the existence of an LCG equivalent to a given CFG can also be provided by methods from section 2. Let G be the BCG derivable from Γ with the initial type assignment resulting from I, the latter mapping being defined according to (15), (16). We have shown $L(G) = L(\Gamma)$. By G' we denote the LCG which results from G by replacing **B** by **L**. Clearly, $L(\Gamma) = L(G) \subseteq L(G')$, since **L** is an extension of **B**. On the other hand, $L(G') \subseteq L(G(\Gamma)) = L(\Gamma)$, since $\mathbf{L}(R^\Gamma)$ is an extension of **L**. So, $L(G') = L(\Gamma)$.

LP is a conservative extension of **L**, that means, the same product-free sequents are derivable in **LP** and **L**. As a consequence, we obtain the equivalence of LCG's and CFG's. However, production rules (22) appearing in the CFG Γ, equivalent to a given LCG G and constructed by the Pentus method, may contain types with product. If we transformed Γ into a BCG with the aid of type transformations described in section 2, we would obtain an "ugly BCG", employing types with product, treated as atomic types in derivation procedures. To construct a natural BCG equivalent to a given LCG we should prove a variant of the BR-lemma for **L**. That is our main goal in this section.

Unfortunately, the Pentus way of establishing the BR-lemma cannot directly be adapted to **L**. The matter is tightly connected with interpolation, which we are to explain now. 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 like (21). Let $\mathbf{LP} \vdash XYZ \to a$ with $Y \neq \Lambda$. The type y is called *an interpolant of string Y with respect to the latter context*, if the following conditions are satisfied:

(26) $\mathbf{LP} \vdash Y \to y$ and $\mathbf{LP} \vdash XyZ \to a$,
(27) $\rho(p,y) \leq \min(\rho(p,Y), \rho(p, XZ \to a))$, for any atomic type p.

As shown in [20], interpolants in the above sense exist for all strings Y satisfying the required assumption. The proof of the BR-lemma given in [19] heavily uses this interpolation property of LP. Actually, the type d in the BR-lemma is an interpolant of string bc in the context $\mathrm{LP} \vdash YbcZ \rightarrow a$. Yet, for the case of L, one cannot proceed this way. Consider the following example:

(28) $\mathrm{L} \vdash pqr \rightarrow (s/pqr)\backslash s$,

which holds by (Ax), (/1) and (\2). Let y be an interpolant of string qr with respect to (28). By (27), we obtain $\rho(q,y), \rho(r,y) \leq 1$, and $\rho(t,y) = 0$, for any atomic type t different from q and r. The only types satisfying this constraint are q, r, q/r, r/q, $q\backslash r$, $r\backslash q$, $q*r$, $r*q$, and only $q*r$ fulfils (26), hence $y = q*r$. Therefore, there is no product-free interpolant of string qr with respect to (28). On the other hand, (28) admits product-free binary reductions:

(29) $qr \rightarrow (s/qr)\backslash s$ and $p((s/qr)\backslash s) \rightarrow (s/pqr)\backslash s$,

but type $(s/qr)\backslash s$ is not an interpolant of string qr with respect to (28), although it satisfies the complexity constraint of the BR-lemma. Consequently, the BR-lemma cannot be proven for L by finding an interpolant d of string bc which satisfies the complexity constraint.

We do not know if the BR-lemma holds for L, though we cannot find any counterexample. Fortunately, its weaker version with type a supposed to be atomic can be proven by a modification of Pentus' argument, which will be shown here, and that is enough for the desired equivalence results.

First, the notion of an interpolant must be adjusted to L. By *an interpolant of string Y ($Y \neq \Lambda$) with respect to the context* $\mathrm{L} \vdash XYZ \rightarrow a$ we mean a string $y_1 \ldots y_n$, of product-free types, such that there are nonempty strings $Y_1 \ldots Y_n$, satisfying $Y = Y_1 \ldots Y_n$ and the following conditions:

(30) $\mathrm{L} \vdash Y_i \rightarrow y_i$, for $i = 1, \ldots, n$,
(31) $\mathrm{L} \vdash Xy_1 \ldots y_n Z \rightarrow a$,
(32) $\rho(p,y_i) \leq \min(\rho(p,Y_i), \rho(p, XY_1 \ldots Y_{i-1}Y_{i+1} \ldots Y_n Z \rightarrow a))$, for $i = 1, \ldots, n$,
(33) $\rho(p,y_1 \ldots y_n) \leq \min(\rho(p,Y), \rho(p, XZ \rightarrow a))$,

for all atomic types p. That means, each type y_i is an interpolant of the corresponding string Y_i and type $y_1 * \ldots * y_n$ is an interpolant of string Y with respect to this context in the previous sense. We prove an analogue of the interpolation lemma from [20] (pp. 84-86).

Lemma 7. If $\mathrm{L} \vdash XYZ \rightarrow a$, $Y \neq \Lambda$, then there is an interpolant of string Y with respect to this context.

Proof. We proceed by induction on derivations of $XYZ \rightarrow a$ in L.

If $XYZ \rightarrow a$ 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 with respect to the premise. Rule (/1) must be examined in detail ((\1) is dual).

Let the rule be $TbV \rightarrow a$, $U \rightarrow c \vdash T(b/c)UV \rightarrow a$. 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^* be an interpolant of U_2 with respect to the right premise, and let V^* be an interpolant of V_1 with respect to the left premise. We take U^*V^* 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^* be an interpolant of U_2 with respect to the right premise, and let T^* be an interpolant of T_2b with respect to the left premise. Then, $T^* = Sd$, $T_2 = T'T''$ and type d is an interpolant of $T''b$ with respect to the left premise. We take the string $S(d/U^*)$ as an interpolant of Y.

We have checked all possible cases, which finishes the proof.

Following [19], we introduce some 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 $X \rightarrow a$ is said to be *thin*, if (1) $L \vdash X \rightarrow a$, (2) every type appearing in $X \rightarrow a$ is thin, (3) $\rho(p, X \rightarrow a) \in \{0,2\}$, for any atomic type p.

Lemma 8 ([19]). Let $a_1 \ldots a_n \rightarrow a_{n+1}$, $n \geq 2$, be a thin sequent. Then, for some $2 \leq k \leq n$, $\pi(a_k) \subseteq \pi(a_{k-1}) \cup \pi(a_{k+1})$.

Proof. Actually, we need not prove this lemma which immediately follows from lemma 4 in [19] establishing the same for **LP**, due to the fact that **LP** is a conservative extension of **L**. It is, however, noteworthy this the only place Pentus [19] essentially uses an interpretation of **LP** in a free group in the following sense. Consider the free group generated by atomic types. Define $g(a)$ by setting: $g(p) = p$, $g(a*b) = g(a)g(b)$, $g(a/b) = g(a)g(b)^{-1}$, $g(a\backslash b) = g(a)^{-1}g(b)$. Then, sequent $a_1 \ldots a_n \rightarrow b$ is derivable in **LP** only if $g(a_1)\ldots g(a_n) = g(b)$ in the free group.

The next lemma is, actually, the BR-lemma for thin sequents with an atomic succedent. In the proof we use the fact that each sequent derivable in **L** must contain an even number of occurrences of any atomic type.

Lemma 9. If $a_1 \ldots a_n \rightarrow p$, $n \geq 2$, is a thin sequent such that $a_i \in Tp_m(P)$, for all $i = 1, \ldots, n$, and $p \in P$, then there are a number $1 \leq k < n$ and type $b \in Tp_m(P)$ such that $L \vdash a_ka_{k+1} \rightarrow b$ and $L \vdash a_1 \ldots a_{k-1}ba_{k+2} \ldots a_n \rightarrow p$.

Proof. The proof applies similar tools as that of lemma 6 from [19], but case (2) below is treated in a different way. Also, we need additional elimination of "long" interpolants. If $n = 2$, then type $b = p$ fulfils the thesis. Assume $n > 2$. Let k be the number satisfying the thesis of lemma 8. Two cases are to be considered.

(1) $k < n$. Then $\pi(a_k) \subseteq \pi(a_{k-1}) \cup \pi(a_{k+1})$. By $card(K)$ we symbolize the cardinality of the set K. We consider two subcases.

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

(34) $\rho(Y) = card(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + card(\pi(a_k) - \pi(a_{k-1}) \cap \pi(a_k)) =$
$= card(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + card(\pi(a_k) \cap \pi(a_{k+1})) \leq$
$\leq card(\pi(a_{k-1}) - \pi(a_{k-1}) \cap \pi(a_k)) + card(\pi(a_{k-1}) \cap \pi(a_k)) =$
$= card(\pi(a_{k-1})) = \rho(a_{k-1})$,

where the first equality holds by (33), since a_{k-1} and a_k are thin, the second equality by the inclusion established in lemma 8, the inequality by the assumption of case (1a), and the remainder is obvious. Now, either Y is a single type, or $Y = ab$, where a, b are interpolants of a_{k-1} and 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 (32),which yields $\rho(Y) > \rho(a_{k-1})$, contrary to (34). Consequently, Y is a single type which belongs to $Tp_m(P)$, again by (34). We set $b = Y$, and our thesis follows from (30), (31).

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

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

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

Now, $\pi(a_{n-1}) \cap \pi(a_n) \neq \emptyset$; otherwise $a_n = p$, but no sequent $Xp \to p$, such that $X \neq \Lambda$ and p does not occur in X , is derivable in L (easy induction on derivations in L). Consequently:

(36) $\rho(Y) \leq \rho(a_{n-1}) - card(\pi(a_{n-1}) \cap \pi(a_n)) + 1 \leq \rho(a_{n-1})$.

As above, we infer that Y is a single type fulfilling the thesis.

We are ready to prove a version of the BR-lemma for L, restricted to sequents with atomic succedents.

Lemma 10. If $L \vdash X \to p$, 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 $X = YbcZ$, $L \vdash bc \to d$ and $L \vdash YdZ \to p$.

Proof. Let $X \to p$ satisfy the assumptions, $X = a_1 \ldots a_n$. We choose a derivation D of $X \to p$ in L; we assume axioms (Ax) in D use atomic types only. For each atomic type q appearing in D, we form a set P_q containing as many different copies of q, as many occurrences of axiom $q \to q$ are there in D. Next, different occurrences of this axiom are replaced by different formulae $q' \to q'$, $q' \in P_q$, for any atomic type q, which transforms D into a new derivation D'. The final sequent of D' is $a_1' \ldots a_n' \to p'$ which 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 any sequent from D'. Let $b_1, \ldots, b_n$ be interpolants of types $a_1', \ldots, a_n'$, respectively, with respect to the last sequent of D'. One easily sees that $b_1 \ldots b_n \to p'$ is a thin sequent. By lemma 9, we find $1 \le k \le n$ and type $b' \in Tp_m(P')$ such that $\mathrm{L} \vdash b_k b_{k+1} \to b'$ as well as $\mathrm{L} \vdash b_1 \ldots b_{k-1} b' b_{k+2} \ldots b_n \to p'$, where P' denotes the join of all sets P_q constructed above (we use $\rho(a_i) = \rho(a_i') \ge \rho(b_i)$, for $i = 1, \ldots, n$). Now, substitute again q for every $q' \in P_q$, for any atomic type q, in the two sequents from the preceding sentence. Since L is closed under substitution, we obtain $\mathrm{L} \vdash c_k c_{k+1} \to d$ and $\mathrm{L} \vdash c_1 \ldots c_{k-1} d c_{k+2} \ldots c_n \to p$, where $d \in Tp_m(P)$. It remains to show $\mathrm{L} \vdash a_i \to c_i$, for $i = 1, \ldots, n$. That follows from $\mathrm{L} \vdash a_i' \to b_i$ by the latter substitution. Since L is closed under (CUT), the thesis is true.

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 these types (we add s_G to P if it does not appear in I_G), $s_\Gamma = s_G$, and the production rules of Γ are (22) (binary rules only) and (23) (L replaces LP). By lemma 10, since L is closed under (CUT) and admits no derivable sequents $a \to p$ with $a \ne p$ (that eliminates unary rules (22)), we eventually obtain:

Theorem 1. If G is an LCG, and Γ is the CFG constructed from G in the way described above, then $L(G) = L(\Gamma)$.

4. Main construction and final comments

In this section we construct, for any LCG G, an equivalent BCG G' in the natural way. That means, the initial type assignment of G' associates with each $v \in V_{G'} = V_G$ a finite collection of types b such that, for some $a \in I_G(v)$, $\mathrm{L} \vdash a \to b$ (so, G' is a natural BCG equivalent to the LCG G in the sense of section 1). To reach the goal we join main constructions from the preceding sections.

Let G be an LCG, and let Γ be the CFG equivalent to G which is referred to in theorem 1. Since the approach of section 2 relies on the assumption nonterminal symbols of a CFG are atomic types, we need a slight modification of Γ. Namely, for each non-atomic type $a \in N_\Gamma$, we introduce a new atomic type p_a, and by N we denote the set of all atomic types from N_Γ and all new types p_a. The modified CFG Γ' is defined as follows. Its terminal symbols and initial symbol are those of Γ, while N is its set of nonterminal symbols. The production rules of Γ' are obtained from those of Γ by replacing each non-atomic type a by p_a. Clearly, nonterminal symbols of Γ' are in a one-one correspondence with nonterminal symbols of Γ, and production rules of both grammars are the same up to this correspondence, which yields $L(\Gamma') = L(\Gamma)$.

Now, let $G^\#$ be the BCG derivable from Γ' according to the construction from section 2 (before lemma 4). We know that $L(G^\#) = L(\Gamma')$. We need an auxiliary notion. *A substitution* is a mapping σ of the set of atomic types into the set of product-free types. Each substitution σ is uniquely extended to a mapping defined on the set of all product-free types by the clauses:

(37) $\sigma(a/b) = \sigma(a)/\sigma(b)$, $\sigma(a\backslash b) = \sigma(a)\backslash\sigma(b)$.

For any BCG H and any substitution σ such that $\sigma(s_H) = s_H$, the BCG $\sigma(H)$ is defined by setting: $V_{\sigma(H)} = V_H$, $I_{\sigma(H)}(v) = \{\sigma(a) : a \in I_H(v)\}$, $s_{\sigma(H)} = s_H$. Clearly:

(38) $L(H) \subseteq L(\sigma(H))$,

since **B** is closed under substitution, that means:

(39) $\mathbf{B} \vdash a_1 \ldots a_n \to b$ implies $\mathbf{B} \vdash \sigma(a_1) \ldots \sigma(a_n) \to \sigma(b)$,

for all types $a_1, \ldots, a_n, b$.

We define $G' = \sigma(G^\#)$, where the substitution σ is given by:

(40) $\sigma(p_a) = a$, for all non-atomic types $a \in N_\Gamma$,
$\sigma(q) = q$, for all atomic types $q \in N_\Gamma$.

Of course, G' is well defined, because the principal type of $G^\#$ equals $s_{\Gamma'}$, and we have $s_{\Gamma'} = s_\Gamma = s_G$, hence $\sigma(s_{\Gamma'}) = s_{\Gamma'}$. By (38), we obtain $L(G^\#) \subseteq L(G')$, and consequently, $L(G) \subseteq L(G')$, since $L(G) = L(\Gamma) = L(\Gamma') = L(G^\#)$. We show:

(41) for any $b \in I_{G'}(v)$, there is $a \in I_G(v)$ such that $\mathbf{L} \vdash a \to b$,

for all $v \in V_{G'}$ (clearly, $V_{G'} = V_G$). Let $b \in I_{G'}(v)$. Then, $b = \sigma(b')$, for some type $b' \in I_{G^\#}(v)$. Since $G^\#$ is derivable from Γ', then there exists an atomic type $p \in N$ such that $p \Rightarrow v$ is in $R_{\Gamma'}$ and $\mathbf{L}(R^{\Gamma'}) \vdash p \to b'$. The following fact:

(42) $\mathbf{L}(R) \vdash a_1 \ldots a_n \to a$ implies $\mathbf{L}(\sigma(R)) \vdash \sigma(a_1) \ldots \sigma(a_n) \to \sigma(b)$,

where $\sigma(R)$ results from replacing each type a in formulae from R by $\sigma(a)$, can be proved by an easy induction on derivations in $\mathbf{L}(R)$ (with (CUT), without (R.Ax)). Applying (42) to our argument, we obtain:

(43) $\mathbf{L}(\sigma(R^{\Gamma'})) \vdash \sigma(p) \to \sigma(b')$.

Now, by definitions of Γ' and σ:

(44) $p \Rightarrow v$ is in $R_{\Gamma'}$ if, and only if, $\sigma(p) \Rightarrow v$ is in R_Γ,
(45) $s \Rightarrow qr$ is in $R_{\Gamma'}$ if, and only if, $\sigma(s) \Rightarrow \sigma(q)\sigma(r)$ is in R_Γ.

Denote $a = \sigma(p)$. By (44) and the definition of Γ, we obtain $a \in I_G(v)$. By (45) and the definition of Γ again, each formula from $\sigma(R^{\Gamma'})$ is derivable in **L**, hence each formula derivable in $\mathbf{L}(\sigma(R^{\Gamma'}))$ is also derivable in **L**. Therefore, (41) follows from (43), using $b = \sigma(b')$.

According to (41), G' is constructed from G in the natural way. We have already shown $L(G) \subseteq L(G')$. The converse inclusion is an easy consequence of (41). For assume $v_1 \ldots v_n \in L(G')$ ($v_i \in V_G$). Then, $\mathbf{B} \vdash b_1 \ldots b_n \to s_G$, for some $b_i \in I_{G'}(v_i)$, $i = 1, \ldots, n$, which implies $\mathbf{L} \vdash b_1 \ldots b_n \to s_G$, for these types. By (41), $\mathbf{L} \vdash a_i \to b_i$, for some $a_i \in I_G(v_i)$, $i = 1, \ldots, n$. So, $\mathbf{L} \vdash a_1 \ldots a_n \to s_G$, by (CUT), which yields $v_1 \ldots v_n \in L(G)$. We have proven $L(G') = L(G)$. As a result, we obtain our main theorem:

Theorem 2. For any LCG G, there is a natural BCG G' equivalent to G.

To explain the construction of G' we recall the construction steps for $I_{G'}$ without referring to auxiliary grammars Γ, Γ' and $G^{\#}$. Fix an LCG G. We consider the set $Tp_m(P)$, where m is the maximal complexity of types appearing in I_G, and P contains s_G and all atomic subtypes of the latter types. The mapping I, of $Tp_m(P)$ into the powerset of $Tp(P)$, is the join of two mappings I_1 and I_2 defined as follows:

(46) $s_G \in I_1(s_G)$,
(47) if $\mathrm{L} \vdash ab \to c$ then $(a\backslash c), ((a\backslash d)/(c\backslash d)) \in I_1(b)$,
(48) if $a \in I_1(b)$ then $(a/(c\backslash b)) \in I_2(c)$,

for all $a,b,c \in Tp_m(P)$. By (9)-(12), if $b \in I(a)$ then $\mathrm{L} \vdash a \to b$. Now, $I_{G'}(v)$ is defined as the join of all sets $I(a)$, for $a \in I_G(v)$.

The equality $L(G) = L(G')$ will remain true, if one adds to $I_{G'}(v)$ other types b satisfying (41), for instance, all types from $I_G(v)$ (thus, G' can be constructed as an expansion of G). An interesting open problem is to characterize minimal expansions G' of G which fulfil $L(G) = L(G')$.

The number of types involved in (46)-(48) polynomially depends on $card(Tp_m(P))$, the latter being majorized by a function polynomial in $card(P)$ and exponential in m. That is not quite bad, since m need not be expected very large for grammars used in linguistics; a main advantage of LCG's, as compared with BCG's, is the possibility of reducing the size of types appearing in the initial type assignment, as more complex types can be derived from simpler ones by the machinery of L. However, (47) appeals to a decision algorithm for L. As far, as we know, a polynomial algorithm has been found only for the fragment of L restricted to types of order at most 2 (Aarts [1]).

The practical efficiency of both constructions, i.e. that of a CFG Γ and that of a BCG G', is nonetheless essentially weakened by the fact that binary rules $\mathrm{L} \vdash ab \to c$, for $a,b,c \in Tp_m(P)$, are not given in an explicit way. Linguists certainly need a transparent axiomatization of these rules, which will also yield a transparent description of type transformations leading from I_G to $I_{G'}$. Maybe, such an axiomatization can be found by close examining the interpolation lemma (lemma 7), but one cannot exclude the invention of an essentially different proof of the equivalence in question be necessary.

Another reason for seeking an equivalence proof different from the Pentus-style argument, presented in section 3, is that the latter cannot be applied to axiomatic extensions of L. We have seen in section 2 that systems L(R), where R consists of production rules of a CFG (up to the direction of arrows), are quite useful in grammar transformations (for a further discussion of L(R)'s, see [6], [9]). We conjecture categorial grammars based on these systems (with R restricted to context-free formulae) are context-free, but this conjecture cannot be proven by the methods of section 3. For neither lemma 8 (using an interpretation of L in a free group), nor the transition to thin sequents, exploited in lemmas 8, 9 and 10, remains plausible for axiomatic extensions of L.

The argument also fails for the system **LP*** (the Lambek Calculus with Permutation) which results from enriching **LP** with the rule:

(PER) $XabY \to c \vdash XbaY \to c$,

characteristic of semantic systems studied in [3], [4], [5] and Girard's linear logics (see [5], [20]). The counterexample is:

(49) **LP*** $\vdash (p/q)/r,(r/s)/t,(t*q)/u \to (p/s)/u$;

the sequent in (49) is thin, but each interpolant of any two antecedent types (not necessarily adjoint) must contain four atomic subtypes, which exceeds $m=3$ (a product-free example can also be produced). Many properties of **LP*** are, however, quite similar to those of **LP**, hence one may expect a modification of the Pentus method will prove that **LP***-languages equal permutation closures of context-free languages (a problem discussed in [4], [9]).

As observed by Pentus [19], the equivalence of **LP1**-grammars and CFG's can be established in the same way as for **LP**-grammars. Also, our construction of a natural BCG equivalent to a given LCG can easily be adapted to the case of **L1**. Alternatively, one may follow the line of [11], where **L1**-derivability has been reduced to **L**-derivability. We recall this reduction for the case of **LP1** and **LP**. For any type a, one defines two finite sets $A(a)$ *and* $S(a)$, of types, such that **LP1** $\vdash a \to b$, for all $b \in A(a)$, and **LP1** $\vdash b \to a$, for all $b \in S(a)$, by the following recursion:

(50) $A(p) = S(p) = \{p\}$, for atomic types p,
$A(a*b) = \{c*d : c \in A(a), d \in A(b)\}$, $S(a*b) = \{c*d : c \in S(a), d \in S(b)\}$,
$A(a/b) = \{c/d : c \in A(a), d \in S(b)\} \cup C(a/b)$,
$A(a\backslash b) = \{c\backslash d : c \in S(a), d \in A(b)\} \cup C(a\backslash b)$,
$S(a/b) = \{c/d : c \in S(a), d \in A(b)\}$, $S(a\backslash b) = \{c\backslash d : c \in A(a), d \in S(b)\}$,
$C(a/b) =$ [if **LP1** $\vdash \Lambda \to b$ **then** $A(a)$ **else** $\emptyset$],
$C(a\backslash b) =$ [if **LP1** $\vdash \Lambda \to a$ **then** $A(b)$ **else** $\emptyset$].

By induction on derivations, one proves **LP1** $\vdash a_1 \ldots a_n \to b$ if, and only if, there exist types $c_i \in A(a_i)$, $i = 1,\ldots,n$, and $d \in S(b)$ such that **LP** $\vdash c_1 \ldots c_n \to d$, for $n > 0$, and the same holds for **L1** versus **L**. Consequently, each grammar based on **LP1** (resp. on **L1**) can effectively be transformed into an equivalent grammar based on **LP** (resp. on **L**), and to the latter one applies the methods considered above in order to find an equivalent CFG or BCG.

References

[1] Aarts, E., Proving Theorems of the Lambek Calculus of order 2 in Polynomial Time, *ILLC Prepublication Series* CT-93-05, University of Amsterdam, 1993.

[2] Bar-Hillel, Y., C. Gaifman & E. Shamir, On categorial and phrase structure grammars, *Bull. Res. Council Israel* F9 (1960), 155-166.

[3] van Benthem, J., *Essays in Logical Semantics*, D. Reidel, Dordrecht, 1986.

[4] van Benthem, J., The Lambek Calculus, in: [18].

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

[6] Buszkowski, W., Some Decision Problems in the Theory of Syntactic Categories, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 28 (1982), 539-548.

[7] Buszkowski, W., The Equivalence of Unidirectional Lambek Categorial Grammars and Context-Free Grammars, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 31 (1985), 369-384.

[8] Buszkowski, W., Generative Capacity of the Nonassociative Lambek Calculus, *Bull. Pol. Acad. Scie. Math.* 34 (1986), 507-516.

[9] Buszkowski, W., Generative Power of Categorial Grammars, in: [18].

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

[11] Buszkowski, W., On Generative Capacity of the Lambek Calculus, in: J. van Eijck (ed.), *Logics in AI*, LNCS, Springer, Berlin, 1991.

[12] Chomsky, N., Formal Properties of Grammars, in: R. D. Luce et al. (ed.),*Handbook of Mathematical Psychology*, vol.2, J. Wiley, New York, 1963.

[13] Cohen, J. M., The Equivalence of Two Concepts of Categorial Grammar, *Information and Control* 10 (1967), 475-484.

[14] Gladkij, A. N., *Formal grammars and languages* (in Russian), Nauka, Moscow, 1973.

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

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

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

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

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

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

[21] Sánchez Valencia, V., *Studies on Natural Logic and Categorial Grammar*, Faculty of Mathematics and Computer Science, University of Amsterdam, 1991.

The ILLC Prepublication Series

CL-91-03 Hub Prüst, Remko Scha, Martin van den Berg — A Formal Discourse Grammar tackling Verb Phrase Anaphora

Other Prepublications

X-91-01 Alexander Chagrov, Michael Zakharyaschev — The Disjunction Property of Intermediate Propositional Logics
X-91-02 Alexander Chagrov, Michael Zakharyaschev — On the Undecidability of the Disjunction Property of Intermediate Propositional Logics
X-91-03 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic
X-91-04 K.N. Ignatiev — Partial Conservativity and Modal Logics
X-91-05 Johan van Benthem — Temporal Logic
X-91-06 — Annual Report 1990
X-91-07 A.S. Troelstra — Lectures on Linear Logic, Errata and Supplement
X-91-08 Giorgie Dzhaparidze — Logic of Tolerance
X-91-09 L.D. Beklemishev — On Bimodal Provability Logics for Π_1-axiomatized Extensions of Arithmetical Theories
X-91-10 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice
X-91-11 Michael Zakharyaschev — Canonical Formulas for K4. Part I: Basic Results
X-91-12 Herman Hendriks — Flexibele Categoriale Syntaxis en Semantiek: de proefschriften van Frans Zwarts en Michael Moortgat
X-91-13 Max I. Kanovich — The Multiplicative Fragment of Linear Logic is NP-Complete
X-91-14 Max I. Kanovich — The Horn Fragment of Linear Logic is NP-Complete
X-91-15 V. Yu. Shavrukov — Subalgebras of Diagonalizable Algebras of Theories containing Arithmetic, revised version
X-91-16 V.G. Kanovei — Undecidable Hypotheses in Edward Nelson's Internal Set Theory
X-91-17 Michiel van Lambalgen — Independence, Randomness and the Axiom of Choice, Revised Version
X-91-18 Giovanna Cepparello — New Semantics for Predicate Modal Logic: an Analysis from a standard point of view
X-91-19 Papers presented at the Provability Interpretability Arithmetic Conference, 24-31 Aug. 1991, Dept. of Phil., Utrecht University
Annual Report 1991

1992 *Logic, Semantics and Philosophy of Langauge*

LP-92-01 Víctor Sánchez Valencia — Lambek Grammar: an Information-based Categorial Grammar
LP-92-02 Patrick Blackburn — Modal Logic and Attribute Value Structures
LP-92-03 Szabolcs Mikulás — The Completeness of the Lambek Calculus with respect to Relational Semantics
LP-92-04 Paul Dekker — An Update Semantics for Dynamic Predicate Logic
LP-92-05 David I. Beaver — The Kinematics of Presupposition
LP-92-06 Patrick Blackburn, Edith Spaan — A Modal Perspective on the Computational Complexity of Attribute Value Grammar
LP-92-07 Jeroen Groenendijk, Martin Stokhof — A Note on Interrogatives and Adverbs of Quantification
LP-92-08 Maarten de Rijke — A System of Dynamic Modal Logic
LP-92-09 Johan van Benthem — Quantifiers in the world of Types
LP-92-10 Maarten de Rijke — Meeting Some Neighbours (a dynamic modal logic meets theories of change and knowledge representation)
LP-92-11 Johan van Benthem — A note on Dynamic Arrow Logic
LP-92-12 Heinrich Wansing — Sequent Caluli for Normal Modal Propositional Logics
LP-92-13 Dag Westerståhl — Iterated Quantifiers
LP-92-14 Jeroen Groenendijk, Martin Stokhof — Interrogatives and Adverbs of Quantification

Mathematical Logic and Foundations

ML-92-01 A.S. Troelstra — Comparing the theory of Representations and Constructive Mathematics
ML-92-02 Dmitrij P. Skvortsov, Valentin B. Shehtman — Maximal Kripke-type Semantics for Modal and Superintuitionistic Predicate Logics
ML-92-03 Zoran Marković — On the Structure of Kripke Models of Heyting Arithmetic
ML-92-04 Dimiter Vakarelov — A Modal Theory of Arrows, Arrow Logics I
ML-92-05 Domenico Zambella — Shavrukov's Theorem on the Subalgebras of Diagonalizable Algebras for Theories containing $I\Delta_0$ + EXP
ML-92-06 D.M. Gabbay, Valentin B. Shehtman — Undecidability of Modal and Intermediate First-Order Logics with Two Individual Variables
ML-92-07 Harold Schellinx — How to Broaden your Horizon
ML-92-08 Raymond Hoofman — Information Systems as Coalgebras
ML-92-09 A.S. Troelstra — Realizability
ML-92-10 V.Yu. Shavrukov — A Smart Child of Peano's

Compution and Complexity Theory

CT-92-01 Erik de Haas, Peter van Emde Boas — Object Oriented Application Flow Graphs and their Semantics
CT-92-02 Karen L. Kwast, Sieger van Denneheuvel — Weak Equivalence: Theory and Applications
CT-92-03 Krzysztof R. Apt, Kees Doets — A new Definition of SLDNF-resolution

Other Prepublications

X-92-01 Heinrich Wansing — The Logic of Information Structures
X-92-02 Konstantin N. Ignatiev — The Closed Fragment of Dzhaparidze's Polymodal Logic and the Logic of Σ_1 conservativity
X-92-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions, revised version
X-92-04 Johan van Benthem — Modeling the Kinematics of Meaning
X-92-05 Erik de Haas, Peter van Emde Boas — Object Oriented Application Flow Graphs and their Semantics, revised version

1993 *Logic, Semantics and Philosophy of Langauge*

LP-93-01 Martijn Spaan — Parallel Quantification
LP-93-02 Makoto Kanazawa — Dynamic Generalized Quantifiers and Monotonicity
LP-93-03 Nikolai Pankrat'ev — Completeness of the Lambek Calculus with respect to Relativized Relational Semantics
LP-93-04 Jacques van Leeuwen — Identity, Quarrelling with an Unproblematic Notion
LP-93-05 Jaap van der Does — Sums and Quantifiers
LP-93-06 Paul Dekker — Updates in Dynamic Semantics
LP-93-07 Wojciech Buszkowski — On the Equivalence of Lambek Categorial Grammars and Basic Categorial Grammars

Mathematical Logic and Foundations

ML-93-01 Maciej Kandulski — Commutative Lambek Categorial Grammars
ML-93-02 Johan van Benthem, Natasha Alechina — Modal Quantification over Structured Domains
ML-93-03 Mati Pentus — The Conjoinablity Relation in Lambek Calculus and Linear Logic
ML-93-04 Andreja Prijatelj — Bounded Contraction and Many-Valued Semantics
ML-93-05 Raymond Hoofman, Harold Schellinx — Models of the Untyped λ-calculus in Semi Cartesian Closed Categories
ML-93-06 J. Zashev — Categorial Generalization of Algebraic Recursion Theory
ML-93-07 A.V. Chagrov, L.A. Chagrova — Algorithmic Problems Concerning First-Order Definability of Modal Formulas on the Class of All Finite Frames
ML-93-08 Raymond Hoofman, Ieke Moerdijk — Remarks on the Theory of Semi-Functors
ML-93-09 A.S. Troelstra — Natural Deduction for Intuitionistic Linear Logic
ML-93-10 Vincent Danos, Jean-Baptiste Joinet, Harold Schellinx — The Structure of Exponentials: Uncovering the Dynamics of Linear Logic Proofs
ML-93-11 Lex Hendriks — Inventory of Fragments and Exact Models in Intuitionistic Propositional Logic

Compution and Complexity Theory

CT-93-01 Marianne Kalsbeek — The Vanilla Meta-Interpreter for Definite Logic Programs and Ambivalent Syntax
CT-93-02 Sophie Fischer — A Note on the Complexity of Local Search Problems
CT-93-03 Johan van Benthem, Jan Bergstra — Logic of Transition Systems
CT-93-04 Karen L. Kwast, Sieger van Denneheuvel — The Meaning of Duplicates in the Relational Database Model
CT-93-05 Erik Aarts — Proving Theorems of the Lambek Calculus of Order 2 in Polynomial Time

Other Prepublications

X-93-01 Paul Dekker — Existential Disclosure, revised version
X-93-02 Maarten de Rijke — What is Modal Logic?
X-93-03 Michiel Leezenberg — Gorani Influence on Central Kurdish: Substratum or Prestige Borrowing
X-93-04 A.S. Troelstra (editor) — Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, Corrections to the First Edition
X-93-05 A.S. Troelstra (editor) — Metamathematical Investigation of Intuitionistic Arithmetic and Analysis, Second, corrected Edition
X-93-06 Michael Zakharyashev — Canonical Formulas for K4. Part II: Cofinal Subframe Logics