Institute for Logic, Language and Computation

COMMUTATIVE LAMBEK CATEGORIAL GRAMMARS

Maciej Kandulski

ILLC Prepublication Series
for Mathematical Logic and Foundations ML-93-01

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

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

COMMUTATIVE LAMBEK CATEGORIAL GRAMMARS

Maciej Kandulski
Institute of Mathematics, Adam Mickiewicz University
Poznań, Poland

ILLC Prepublications
for Mathematical Logic and Foundations
ISSN 0928-3315
Coordinating editor: Dick de Jongh

received January 1993

Commutative Lambek categorial grammars

Maciej Kandulski
Institute of Mathematics, Adam Mickiewicz University
ul. Matejki 48/49, 60-769 Poznań, Poland
mkandu@plpuam11.bitnet

1 Introduction

This paper is devoted to the problem of establishing the generative power of categorial grammars (CGs) based on the nonassociative and commutative Lambek calculus with product (NCL). Investigations concerning generative power of CGs, which date from the paper of Bar-Hillel et al. [2], are stimulated by two factors. The first of them is the problem of the generative power of CGs based on the Lambek calculus [19]. This question posed in [2] resulted in a series of papers characterizing classes of languages generated by CGs based on subsystems of the Lambek calculus (see [6, 7, 9, 17, 18, 28]) and was recently solved by Pentus [25]. On the other hand, in order to describe in a better way some phenomena of natural languages the shape of type reduction systems in CGs was modified so as to obtain more flexible versions of the formalism; the reader will find a detailed description of this and other relevant problems in [5]. Modifications included, among others, a relaxation of the structure of antecedents of reduction formulas. Those changes used to be introduced via endowing a type reduction system with structural rules of permutation, contraction and monotonicity performed as it is the case in Gentzen systems **LJ** and **LK** [14]. The introduction of structural rules implies changes in the formal status of antecedents of formulas: if in the case of basic Lambek calculus (associative and noncommutative) antecedents can be viewed as finite sequences of types then for a commutative calculus the proper model for an antecedent is that of a multiset, if we work with a nonassociative calculus then antecedents are bracketed strings of types (trees) and if a calculus is both commutative and nonassociative then antecedents can be considered as mobiles, see [22]. Generative power of categorial grammars enriched with structural rules of permutation and contraction was studied by van Benthem in [3] and [4] . Let us observe that affixing permutation to the Lambek calculus results in a rather undesired effect: one must accept sentences with a completely free word order. Although there is some linguistic evidence in favour of such general relaxation, see for example [1], but still this global effect caused by permutation seems to be too strong from the point of view of a linguist. As a way out of this situation one can considered nonassociative calculi [20]: by the presence of the permutation they admit commutativity only within phrase structures. A local, restricted introduction of structural rules can also be performed by means of structural modalities, see [12] and [23]. This method is similar to the procedure of retrieving structural rules in Linear Logic, cf. [27].

In this paper we prove that the class of languages generated by CGs based on NCL coincides with the class of CF–languages of finite degree whose rules are closed with respect to permutation ($\overline{\text{CF}}$–languages). To establish the inclusion of the class of NCL-languages in the class of $\overline{\text{CF}}$–languages we employ the notion of normal form of derivation which already proved useful in investigations concerning generative power of categorial grammars, see [7] and [17]. The proof of the converse inclusion makes use of an algebraic characterization of phrase languages generated by classical (Ajdukiewicz) CGs and CF-grammars, see [7] and [26].

The presented work is a unified version of two articles submitted to *Mathematical Logic Quarterly*, formerly *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik.* A part of this paper is a result of the author's stay in the Institute of Language, Logic and Information at the University of Amsterdam as a TEMPUS guest in May and June 1992. The author would like to thank the staff of the Institute for hospitality and stimulating atmosphere of work.

2 The nonassociative and commutative Lambek calculus NCL

Let Pr be an infinite but countable set of *primitive types* and let $/$, $\backslash$ and $\cdot$ denote three binary operations called *left division, right division* and *product* respectively. The set TP, of *types (with product)* is the smallest set containing Pr and closed with respect to $/$, $\backslash$ and $\cdot$. By Tp we denote the product-free part of TP. We call x/y or $y\backslash x$ (resp. $x \cdot y$) a *functorial* (resp. *product*) type. The symbol $c(x)$ is used to denote the *complexity* of type x, i.e. the number of occurrences of $/$, $\backslash$ and $\cdot$ in x. The set $\text{sub}(x)$ of *subtypes* of x is defined by the following induction: (i) $x \in \text{sub}(x)$, (ii) if $x = y/z$ or $x = z\backslash y$ or $x = y \cdot z$, then $y \in \text{sub}(x)$ and $z \in \text{sub}(x)$. For $A \subseteq \text{TP}$ we put $\text{sub}(A) = \bigcup_{x \in A}\text{sub}(x)$. By the set of *bracketed strings of types* we mean the smallest set defined inductively as follows: (i) $\text{TP} \subseteq \text{BSTP}$, (ii) if $X \in \text{BSTP}$ and $Y \in \text{BSTP}$ then $(X, Y) \in \text{BSTP}$. The product-free part of BSTP will be denoted by BSTp.

The nonassociative Ajdukiewicz calculus (with product) NA is a formal system whose formulas are of the form $X \rightarrow x$, where $x \in \text{BSTP}$ and $x \in \text{TP}$. NA has one axiom scheme

(A0) $\quad x \rightarrow x$, where $x \in \text{TP}$,

and the following rules of inference:

$$\text{(A)} \quad \frac{X \rightarrow x/y \qquad Y \rightarrow y}{(X, Y) \rightarrow x} \qquad\qquad (\overline{\text{A}}) \quad \frac{X \rightarrow y\backslash x \qquad Y \rightarrow y}{(Y, X) \rightarrow x}$$

$$\text{(PR)} \quad \frac{X \rightarrow x \qquad Y \rightarrow y}{(X, Y) \rightarrow x \cdot y}$$

The nonassociative and commutative Ajdukiewicz calculus (with product) NCA arises from NA by adding the following (structural) rule of permutation:

$$(\text{Perm})\ \frac{(X,Y)\to z}{(Y,X)\to z}, \quad \text{where } X,Y\in \text{BSTP},\ z\in \text{TP}.$$

As $\vdash_{\text{NCA}} x/y \to y\backslash x$ and $\vdash_{\text{NCA}} y\backslash x \to x/y$, there is no need to differentiate between types x/y and $y\backslash x$ in NCA. We will use the slash / to denote the only division in this system. NCA however admits an equivalent axiomatization consisting of (A0), (A), (PR) as well as two additional rules of inference:

$$(\text{A}')\quad \frac{X\to x/y \qquad Y\to y}{(Y,X)\to x} \qquad\qquad (\text{PR}')\ \frac{X\to x \qquad Y\to y}{(Y,X)\to x\cdot y}$$

In order to define the nonassociative and commutative Lambek calculus (with product) NCL we need a certain auxiliary system Ax. Formulas in Ax are of the shape $x\to y$, where $x,y\in$ TP and it admits the following axiom schemata and rules of inference for all $x,y,z\in$ TP:

(A0)	$x\to x$		
(A1)	$(x/y)\cdot y\to x$	(A1′)	$y\cdot(x/y)\to x$
(A2)	$x\to(x\cdot y)/y$	(A2′)	$x\to(y\cdot x)/y$
(A3)	$x\to y/(y/x)$	(A4)	$x\cdot y\to y\cdot x$

$$(\text{R1})\quad \frac{x\to y}{x/z\to y/z} \qquad\qquad (\text{R1}')\quad \frac{x\to y}{z/y\to z/x}$$

$$(\text{R2})\quad \frac{x\to y}{x\cdot z\to y\cdot z} \qquad\qquad (\text{R2}')\quad \frac{x\to y}{z\cdot x\to z\cdot y}$$

To simplify the notation we write $x\to y\in$ Ax instead of $\vdash_{\text{Ax}} x\to y$. The calculus NCL amounts to the system NCA enriched with the following compound rule:

$$(\text{C})\quad \frac{X\to x}{X\to y}, \quad \text{if } x\to y\in \text{Ax and } x\neq y.$$

Product-free versions of all introduced calculi, which employ types from Tp instead of TP and make use of no product axioms and rules, will be denoted NA^{O}, NCA^{O} and NCL^{O} respectively.

Let $x\to y\in$ Ax. We call $x\to y$ an E–*formula* (resp. R–*formula* or O–*formula*) if $c(x)<c(y)$ (resp. $c(x)>c(y)$ or $c(x)=c(y)$). To emphasis the fact that we deal with an E– (resp. R– or O–) formula from Ax we write $x\xrightarrow{\text{E}}y\in$ Ax (resp. $x\xrightarrow{\text{R}}y\in$ Ax or $x\xrightarrow{\text{O}}y\in$ Ax).

The following Lemma is a counterpart of the result obtained for the nonassociative and noncommutative calculi (see [7] and [17]):

Lemma 1. *If* $x\xrightarrow{\text{E}}y\in$ Ax *and* $y\xrightarrow{\text{R}}z\in$ Ax, *where* $x\neq y$, *then there exists* $\overline{y}\in$ TP *such that* $c(\overline{y})<c(y)$ *and* $x\to\overline{y}\in$ Ax *and* $\overline{y}\to z\in$ Ax.

Proof. By induction on $c(x)$ as in [7] or [17]. □

From now on we will use a certain notational convention: Instead of the sequence $x_1\to x_2\in$ Ax, $x_2\to x_3\in$ Ax, ..., $x_{n-1}\to x_n\in$ Ax we will write $x_1\to x_2\to x_3\to\cdots\to x_n\in$

Ax and call $x_1 \to x_2 \to x_3 \to \cdots \to x_n$ a *sequence derivable in* Ax. We reserve the letters $S, T, U, \ldots$ for denoting derivable sequences. If S is a derivable sequence different from the axiom (A0), then by $l(S)$ we denote the *length of S*, defined as the number of arrows in S. Thus, all axioms, except for (A0), have the length 1. Additionally, for a sequence T being an instance of (A0) we put $l(T) = 0$. The notation employing indices E, R or O above the arrow in a formula can in a natural way be extended to the case of derivable sequences. For every derivable sequence $x_1 \to \cdots \to x_n$ the string $I_1 I_2 \ldots I_{n-1}$ is called the *index* of this sequence, if for every $i \in \{1, \ldots, n-1\}$ we have $x_i \xrightarrow{I_i} x_{i+1} \in$ Ax. The interest of Lemma 1 is that it allows to transform a sequence $x \to y \to z$ of index ER into a sequence $x \to \overline{y} \to z$ of any but ER index.

Lemma 2.

(i) *If* $x_1/y_1 \xrightarrow{\mathrm{O}} x_2/y_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n/y_n \in$ Ax, *then* $x_1 \xrightarrow{\mathrm{O}} x_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n \in$ Ax *and* $y_n \xrightarrow{\mathrm{O}} y_{n-1} \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} y_1 \in$ Ax.

(ii) *If* $x_1 \cdot y_1 \xrightarrow{\mathrm{O}} x_2 \cdot y_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n \cdot y_n \in$ Ax, *then* $x_1 \xrightarrow{\mathrm{O}} v_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} v_n \in$ Ax *and* $y_1 \xrightarrow{\mathrm{O}} z_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n \in$ Ax, *where for* $2 \leq i \leq n$ *either* $v_i = x_i$ *and* $z_i = y_i$ *or* $v_i = y_i$ *and* $z_i = x_i$.

Proof. By induction on n. □

The indicated in Lemma 2 sequences whose initial types are x_1 and y_1 will be called the *constituents* of the sequence $x_1/y_1 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n/y_n$ and $x_1 \cdot y_1 \xrightarrow{\mathrm{O}} \cdot \xrightarrow{\mathrm{O}} x_n \cdot y_n$, respectively. By the *reduced form* of a sequence S we mean the sequence obtained from S by dropping all instances of (A0) in S. This definition does not apply to formulas which consist entirely of the axiom (A0). In this case, by the reduced form of $x \to x \to \cdots \to x$ we mean the sequence (formula) $x \to x$. A sequence of O-formulas is said to be *reduced* if all formulas occurring in this sequence are different from (A0).

Lemma 3. *Let S be a derivable sequence of the form* $x_1/y_1 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n/y_n$ *or* $x_1 \cdot y_1 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n \cdot y_n$ *and let T and U be the reduced forms of the constituents of S. Then we have:*

(i) $l(S) \geq l(T) + l(U)$,

(ii) *if the sequence S is reduced, then* $l(S) = l(S) + l(U)$.

Proof. We employ Lemma 2. □

Lemma 4. *If* $x \xrightarrow{\mathrm{E}} y \xrightarrow{\mathrm{O}} z \xrightarrow{\mathrm{R}} v \in$ Ax, *where* $x \neq v$, *then there exists a derivable sequence of one of the indices* ER, OER, ERO *such that its initial and terminal types are x and y respectively.*

Proof. The detailed inductive proof of the lemma is rather long, we have to consider all cases according to the inductive definition of formulas in Ax. Thus, for the sake of brevity we disregard parts of the proof which are dual or similar to those presented previously. First observe that if $y \to z$ is an instance of (A0), then $y = z$ and we can put $x \to y \to v$ as the desired string of index ER. Thus, in our induction we omit this case.

A. Let $x \xrightarrow{\mathrm{E}} y =$ (A2). Then $y = (x \cdot y')/y'$ and $y \xrightarrow{\mathrm{O}} z$ must arise by (R1) or (R1′).

A.1. Assume $y \xrightarrow{O} z = (x \cdot y')/y' \xrightarrow{O} z'/y'$ and $x \cdot y' \xrightarrow{O} z' \in \mathrm{Ax}$. Then $z \xrightarrow{R} v$ must arise by (R1) or (R1′).

A.1.1. $z'/y' \xrightarrow{R} v'/y'$ and $z' \xrightarrow{R} v' \in \mathrm{Ax}$. Then our sequence is of the form $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} z'/y' \xrightarrow{R} v'/y'$ and we have three possibilities:

A.1.1.(a) $x \cdot y' \to z' = x \cdot y' \to y' \cdot x$. Then $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} (y' \cdot x)/y' \xrightarrow{R} v'/y'$ can be replaced by $x \xrightarrow{E} (y' \cdot x)/y' \xrightarrow{R} v'/y'$.

A.1.1.(b) $x \cdot y' \to z' = x \cdot y' \to x \cdot y''$ and $y' \xrightarrow{O} y'' \in \mathrm{Ax}$. Then we can replace $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} (x \cdot y'')/y' \xrightarrow{R} v'/y'$ by $x \xrightarrow{E} (x \cdot y'')/y'' \xrightarrow{R} u'/y'' \xrightarrow{O} v'/y'$.

A.1.1.(c) $x \cdot y' \to z' = x \cdot y' \to x' \cdot y'$ and $x \xrightarrow{O} x' \in \mathrm{Ax}$. Then instead of $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} (x' \cdot y')/y' \xrightarrow{R} v'/y'$ we put $x \xrightarrow{O} x' \xrightarrow{E} (x' \cdot y')/y' \xrightarrow{R} v'/y'$.

A.1.2. $z'/y' \xrightarrow{E} z'/y''$ and $y'' \xrightarrow{E} y' \in \mathrm{Ax}$. Then $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} z'/y' \xrightarrow{R} z'/y''$ can be replaced by $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{R} (x \cdot y')/y'' \xrightarrow{O} z'/y''$.

A.2. Assume $y \xrightarrow{O} z = (x \cdot y')/y' \xrightarrow{O} (x \cdot y')/y''$ and $y'' \xrightarrow{O} y' \in \mathrm{Ax}$. Then the formula $z \xrightarrow{O} v$ must arise by (R1) or (R1′).

A.2.1. $z \xrightarrow{R} v = (x \cdot y')/y'' \xrightarrow{R} v'/y''$ and $x \cdot y' \xrightarrow{R} v' \in \mathrm{Ax}$. Then instead of the sequence $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{R} (x \cdot y')/y'' \xrightarrow{R} v'/y''$ we put $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{R} v'/y' \xrightarrow{O} v'/y''$.

A.2.2. $z \xrightarrow{R} v = (x \cdot y')/y'' \xrightarrow{R} (x \cdot y')/v'$ and $v' \xrightarrow{E} y'' \in \mathrm{Ax}$. Then instead of $x \xrightarrow{E} (x \cdot y')/y' \xrightarrow{O} (x \cdot y')/y'' \xrightarrow{R} (x \cdot y')/v'$ we put $x \xrightarrow{E} (x \cdot y'')/y'' \xrightarrow{R} (x \cdot y'')/v' \xrightarrow{O} (x \cdot y')/v'$.

B. If $x \to y = (\mathrm{A2}')$, then the proof is dual to A.

C. Let $x \xrightarrow{E} y = (\mathrm{A3})$. Then $y = y'/(y'/x)$ and $y \xrightarrow{O} z$ must arise by (R1) or (R1′).

C.1. Assume $y \xrightarrow{O} z = y'/(y'/x) \xrightarrow{O} y''/(y'/x)$ and $y' \xrightarrow{O} y'' \in \mathrm{Ax}$. Then the formula $z \xrightarrow{R} v$ must arise by (R1) or (R1′).

C.1.1. $z \xrightarrow{R} v = y''/(y'/x) \xrightarrow{R} v'/(y'/x)$ and $y'' \xrightarrow{R} v' \in \mathrm{Ax}$. Then instead of $x \xrightarrow{R} y'/(y'/x) \xrightarrow{O} y''/(y'/x) \xrightarrow{R} v'/(y'/x)$ we put $x \xrightarrow{E} y''/(y''/x) \xrightarrow{R} v'/(y''/x) \xrightarrow{O} v'/(y'/x)$.

C.1.2. $z \xrightarrow{R} v = y''/(y'/x) \xrightarrow{R} y''/v'$ and $v' \xrightarrow{E} y' \xrightarrow{E} y'/x \in \mathrm{Ax}$. Then instead of $x \xrightarrow{E} y'/(y'/x) \xrightarrow{O} y''/(y'/x) \xrightarrow{R} y''/v'$ we put $x \xrightarrow{E} y'/(y'/x) \xrightarrow{R} y'/v' \xrightarrow{O} y''/v'$.

C.2. Assume $y \xrightarrow{O} z = y'/(y'/x) \xrightarrow{O} y'/z'$ and $z' \xrightarrow{O} y'/x \in \mathrm{Ax}$. Then the formula $z \xrightarrow{R} v$ must arise by (R1) or (R1′).

C.2.1. $z \xrightarrow{R} v = y'/z' \xrightarrow{R} v'/z'$ and $y' \xrightarrow{R} v' \in \mathrm{Ax}$. Then instead of the sequence $x \xrightarrow{E} y'/(y'/x) \xrightarrow{O} y'/z' \xrightarrow{R} v'/z'$ we put $x \xrightarrow{E} y'/(y'/x) \xrightarrow{R} v'/(y'/x) \xrightarrow{O} v'/z'$.

C.2.2. $z \xrightarrow{R} v = y'/z' \xrightarrow{R} y'/v'$ and $v' \xrightarrow{E} z' \in \mathrm{Ax}$. Thus we have two possibilities:

C.2.2.(a) $z' \xrightarrow{O} y'/x = z''/x \xrightarrow{O} y'/x$ and $z'' \xrightarrow{O} y' \in \mathrm{Ax}$. Then instead of $x \xrightarrow{E}$

$y'/(y'/x) \xrightarrow{\text{O}} y'/(z''/x) \xrightarrow{\text{R}} y'/v'$ we put $x \xrightarrow{\text{E}} z''/(z''/x) \xrightarrow{\text{R}} z''/v' \xrightarrow{\text{O}} y'/v'$.

C.2.2.(b) $z' \xrightarrow{\text{O}} y'/x = y'/z'' \xrightarrow{\text{O}} y'/x$ and $x \xrightarrow{\text{O}} z'' \in \text{Ax}$. Then instead of $x \xrightarrow{\text{E}} y'/(y'/x) \xrightarrow{\text{O}} y'/(y'/z'') \xrightarrow{\text{R}} y'/v'$ we put $x \xrightarrow{\text{O}} z'' \xrightarrow{\text{E}} y'/y'/z'') \xrightarrow{\text{R}} y'/v'$.

D. $x \xrightarrow{\text{E}} y$ arises by (R1). Thus $x \xrightarrow{\text{E}} y = x'/u \xrightarrow{\text{E}} y'/u$ and $x' \xrightarrow{\text{E}} y' \in \text{Ax}$. Then the formula $y \xrightarrow{\text{O}} z$ must arise by (R1) or (R1′).

D.1. Assume $y \xrightarrow{\text{O}} z = y'/u \xrightarrow{\text{O}} z'/u$ and $y' \xrightarrow{\text{O}} z' \in \text{Ax}$. Then the formula $z \xrightarrow{\text{R}} v$ must arise by (R1) or (R1′).

D.1.1. $z \xrightarrow{\text{R}} v = z'/u \xrightarrow{\text{R}} v'/u$ and $z' \xrightarrow{\text{R}} v' \in \text{Ax}$. For $x' \xrightarrow{\text{E}} y' \xrightarrow{\text{O}} z' \xrightarrow{\text{R}} v'$ we employ the inductive assumption and divide the resulting sequence of one of the indices ER, OER or ERO by u.

D.1.2. $z \xrightarrow{\text{R}} v = z'/u \xrightarrow{\text{R}} z'/v'$ and $v' \xrightarrow{\text{E}} u \in \text{Ax}$. Then instead of the sequence $x'/u \xrightarrow{\text{E}} y'/u \xrightarrow{\text{O}} z'/u \xrightarrow{\text{R}} z'/v'$ we put $x'/u \xrightarrow{\text{E}} y'/u \xrightarrow{\text{R}} y'/v \xrightarrow{\text{O}} z'/v'$.

D.2. Assume $y \xrightarrow{\text{O}} z = y'/u \xrightarrow{\text{O}} y'/z'$ and $z' \xrightarrow{\text{O}} u \in \text{Ax}$. Then the formula $z \xrightarrow{\text{R}} v$ must arise by (R1) or (R1′).

D.2.1. $z \xrightarrow{\text{R}} v = y'/z' \xrightarrow{\text{R}} v'/z'$ and $y/ \xrightarrow{\text{R}} v' \in \text{Ax}$. Then instead of the sequence $x'/u \xrightarrow{\text{E}} y'/u \xrightarrow{\text{O}} y'/z' \xrightarrow{\text{R}} v'/z'$ we put $x'/u \xrightarrow{\text{O}} x'/z' \xrightarrow{\text{E}} y'/z' \xrightarrow{\text{R}} v'/z'$.

D.2.2. $z \xrightarrow{\text{R}} v = y'/z' \xrightarrow{\text{R}} y'/v'$ and $v' \xrightarrow{\text{E}} z' \in \text{Ax}$. Then instead of the sequence $x'/u \xrightarrow{\text{E}} y'/u \xrightarrow{\text{O}} y'/z' \xrightarrow{\text{R}} y'/v'$ we put $x'/u \xrightarrow{\text{O}} x'/z' \xrightarrow{\text{E}} y'/z' \xrightarrow{\text{R}} y'/v'$.

E. If $x \xrightarrow{\text{E}} y$ arises by (R1′), then we proceed as in D.

F. $x \xrightarrow{\text{E}} y$ arises by (R2). Thus $x \xrightarrow{\text{E}} y = x' \cdot u \xrightarrow{\text{E}} y' \cdot u$ and $x' \xrightarrow{\text{E}} y' \in \text{Ax}$. The formula $y \xrightarrow{\text{O}} z$ may be either the axiom (A4) or the result of an application of the rule (R2) or (R2′).

F.1. Assume $y \xrightarrow{\text{O}} z = y' \cdot u \xrightarrow{\text{O}} u \cdot y'$. Then we have the following four subcases for the formula $z \xrightarrow{\text{O}} v$:

F.1.1. $z \xrightarrow{\text{R}} v = $ (A1). Then $u = v/y'$ and $x' \cdot (v/y') \xrightarrow{\text{R}} y' \cdot (v/y') \xrightarrow{\text{O}} (v/y') \cdot y' \xrightarrow{\text{R}} v$ can be replaced by $x' \cdot (v/y') \xrightarrow{\text{E}} y' \cdot (v/y') \xrightarrow{\text{R}} v$.

F.1.2. $z \xrightarrow{\text{R}} v = $ (A1′). Then the argument is similar to that presented in F.1.1.

F.1.3. $z \xrightarrow{\text{R}} v = u \cdot y' \xrightarrow{\text{R}} v \cdot y'$ and $u \xrightarrow{\text{R}} v \in \text{Ax}$. Then instead of the sequence $x' \cdot u \xrightarrow{\text{E}} y' \cdot u \xrightarrow{\text{O}} u \cdot y' \xrightarrow{\text{R}} v \cdot y'$ we can write $x' \cdot u \xrightarrow{\text{E}} y' \cdot u \xrightarrow{\text{R}} y' \cdot v \xrightarrow{\text{O}} v \cdot y'$.

F.1.4. $z \xrightarrow{\text{R}} v = u \cdot y' \xrightarrow{\text{R}} u \cdot v'$. Then we proceed as in F.1.3.

F.2. Assume $y \xrightarrow{\text{O}} z = y' \cdot u \xrightarrow{\text{O}} z' \cdot u$ and $y' \xrightarrow{\text{O}} z' \in \text{Ax}$. As in F.1. we have the following four subcases:

F.2.1. $z \xrightarrow{\text{R}} v = $ (A1). Then $z' = v/u$. If $y' \xrightarrow{\text{O}} v/u = y''/u \xrightarrow{\text{O}} v/u$ and $y'' \xrightarrow{\text{O}} v \in \text{Ax}$, then instead of $x' \cdot u \xrightarrow{\text{E}} (y''/u) \cdot u \xrightarrow{\text{O}} (v/u) \cdot u \xrightarrow{\text{R}} v$ we can put $x' \cdot u \xrightarrow{\text{E}} (y''/u) \cdot u \xrightarrow{\text{R}} y'' \xrightarrow{\text{O}} v$.

If $y' \xrightarrow{\mathrm{O}} v/u = v/y'' \xrightarrow{\mathrm{O}} v/u$ and $u \xrightarrow{\mathrm{O}} y'' \in \mathrm{Ax}$, then we repeat the preceding argument.

F.2.2. $z \xrightarrow{\mathrm{R}} v = (\mathrm{A1}')$. We proceed as in F.2.1.

F.2.3. $z \xrightarrow{\mathrm{R}} v = z' \cdot u \xrightarrow{\mathrm{R}} v' \cdot u$ and $z' \xrightarrow{\mathrm{R}} v' \in \mathrm{Ax}$. We employ the inductive assumption for the sequence $x' \xrightarrow{\mathrm{E}} y' \xrightarrow{\mathrm{O}} z' \xrightarrow{\mathrm{R}} v'$ and get a sequence of the index ER, OER or ERO which must be multiplied by u to substitute for $x' \cdot u \xrightarrow{\mathrm{E}} y' \cdot u \xrightarrow{\mathrm{O}} z' \cdot u \xrightarrow{\mathrm{R}} v' \cdot u$.

F.2.4. $z \xrightarrow{\mathrm{R}} v = z' \cdot u \xrightarrow{\mathrm{R}} z' \cdot v'$ and $u \xrightarrow{\mathrm{R}} v' \in \mathrm{Ax}$. Then instead of the sequence $x' \cdot u \xrightarrow{\mathrm{E}} y' \cdot u \xrightarrow{\mathrm{O}} z' \cdot u \xrightarrow{\mathrm{R}} z' \cdot v'$ we can put $x' \cdot u \xrightarrow{\mathrm{E}} y' \cdot u \xrightarrow{\mathrm{R}} y' \cdot v' \xrightarrow{\mathrm{O}} z' \cdot y'$.

F.3. Assume $y \xrightarrow{\mathrm{O}} z = y' \cdot u \xrightarrow{\mathrm{O}} y' \cdot z'$. We proceed as in F.2.

G. If $x \xrightarrow{\mathrm{E}} y$ arises by (R2′), then the proof is similar to that presented in F. □

To simplify the notation we write O^n to denote a group of n consecutive indices O occurring in the index of a certain sequence.

Lemma 5. *Let $x_1 \to \cdots \to x_{n+3}$, where $n \geq 1$ and $x_1 \neq x_{n+3}$, be a derivable sequence of the index* $\mathrm{EO}^n\mathrm{R}$ *and let no formula in this sequence be an instance of* (A0). *Then there exists a sequence $y \to \cdots \to z \in \mathrm{Ax}$ whose index is either O^k, where $k \leq n+2$, or $\mathrm{O}^k\mathrm{EO}^l\mathrm{RO}^m$, where $k+l+m \leq n$ and $k+m \neq 0$, such that $x_1 = y$ and $x_{n+3} = z$.*

Proof. We must take in consideration all forms of formulas being the initial and terminal parts of this sequence. According to the assumption of the lemma the sequence $x_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_{n+2}$ is reduced. Moreover, it is of one of the following forms: either $x_2'/x_2'' \xrightarrow{\mathrm{O}} \cdots\cdots \xrightarrow{\mathrm{O}} x_{n+2}'/x_{n+2}''$ or $x_2'' \cdot x_2'' \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_{n+2}' \cdot x_{n+2}''$. Throughout the proof, instead of the constituents of the above sequence we will employ their reduced forms, but using the same notation as the initial and terminal types of a sequence and its reduced form are the same. It allows us to make use of Lemma 3 when necessary.

A. $x_1 \xrightarrow{\mathrm{E}} x_2 = (\mathrm{A2})$. Then the sequence $x_2 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_{n+2}$ must consist of functorial types.

A.1. $x_{n+2} \xrightarrow{\mathrm{R}} x_{n+3} = v/u \xrightarrow{\mathrm{R}} z/u$ and $v \xrightarrow{\mathrm{R}} z \in \mathrm{Ax}$. Then we have $x \xrightarrow{\mathrm{E}} (x \cdot y)/y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} v/u \xrightarrow{\mathrm{R}} z/u \in \mathrm{Ax}$, and let $x \cdot y \to \cdots \to v$ and $u \to \cdots \to y$ be the reduced forms of the constituents of the O^n–part of our sequence.

A.1.1. If $u \neq y$, then instead of the initial sequence we can put the sequence $x \xrightarrow{\mathrm{E}} (x \cdot y)/y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} v/y \xrightarrow{\mathrm{R}} z/y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/u$ and, by Lemma 3, reduce the number of O–formulas between the E–formula and the R–formula. (After this remark about the way in which we use Lemma 3 we will perform further similar transformations without any comments.)

A.1.2. If $u = y$, then we have to consider the following four subcases:

A.1.2.(a) $v \xrightarrow{\mathrm{R}} z = (\mathrm{A1})$. Then $x \xrightarrow{\mathrm{E}} (x \cdot y)/y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} ((z/v') \cdot v')/y \xrightarrow{\mathrm{R}} z/y \in \mathrm{Ax}$. If $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/v' \in \mathrm{Ax}$ and $y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} v' \in \mathrm{Ax}$, then we replace our sequence by $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/v' \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/y'$, provided $y \neq v'$, and by $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/v'$, provided $y = v'$. If $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} v' \in \mathrm{Ax}$ and $y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z/v' \in \mathrm{Ax}$, then the initial sequence can be replaced by $x \xrightarrow{O} \cdots \xrightarrow{O} v' \xrightarrow{E} (v' \cdot (z/v'))/(z/v') \xrightarrow{R} z/(z/v') \xrightarrow{O} \cdots \xrightarrow{O} z/y$, provided $x \neq v'$, and by $x \xrightarrow{E} (x \cdot (z/x))/(z/x) \xrightarrow{R} z/(z/x) \xrightarrow{O} \cdot \xrightarrow{O} z/y$, provided $x = v'$.

A.1.2.(b) If $v \xrightarrow{R} z = (A1')$, then we proceed as in A.1.2.(a).

A.1.2.(c) $v \xrightarrow{R} z = v' \cdot t \xrightarrow{R} z' \cdot t$ and $v' \xrightarrow{R} z' \in$ Ax. This case is similar to A.1.2.(a) and is left to the reader.

A.1.2.(d) $v \xrightarrow{R} z = t \cdot v' \xrightarrow{R} t \cdot z'$ and $v' \xrightarrow{R} z' \in$ Ax. In this case we proceed as in A.1.2.(c).

A.2. $x_{n+2} \xrightarrow{R} x_{n+3} = u/z \xrightarrow{R} u/v$ and $v \xrightarrow{E} z \in$ Ax. In this case we proceed in a similar way as in A.1.

B. $x_1 \xrightarrow{E} x_2 = (A2')$. The proof is dual to that presented in A.

C. $x_1 \xrightarrow{E} x_2 = (A3)$. The proof is similar to that presented in A.

D. $x_1 \xrightarrow{E} x_2 = x/u \xrightarrow{E} y/u$ and $x \xrightarrow{E} y \in$ Ax. There are two possibilities:

D.1. $x_{n+2} \xrightarrow{R} x_{n+3} = v/t \xrightarrow{R} z/t$ and $v \xrightarrow{R} z \in$ Ax. Let $y \xrightarrow{O} \cdots \xrightarrow{O} v$ and $t \xrightarrow{O} \cdots \xrightarrow{O} u$ be the reduced forms of the constituents of $y/u \xrightarrow{O} \cdots \xrightarrow{O} v/t$.

D.1.1. If $t \neq u$, then we replace the sequence $x/u \xrightarrow{E} y/u \xrightarrow{O} \cdots \xrightarrow{O} v/t \xrightarrow{R} z/t$ by $x/u \xrightarrow{E} y/u \xrightarrow{O} \cdots \xrightarrow{O} v/u \xrightarrow{R} z/u \xrightarrow{O} \cdots \xrightarrow{O} z/t$.

D.1.2. If $t = u$, then we make use of the inductive assumption for the sequence $x \xrightarrow{E} y \xrightarrow{O} \cdots \xrightarrow{O} v \xrightarrow{R} z$ and divide the resulting string by u.

D.2. $x_{n+2} \xrightarrow{R} x_{n+3} = t/z \xrightarrow{R} t/v$ and $v \xrightarrow{E} z \in$ Ax. This case is similar to D.1 and is left to the reader.

E. $x_1 \xrightarrow{E} x_2 = u/x \xrightarrow{R} u/y$ and $y \xrightarrow{R} x \in$ Ax. In this case we proceed similar as in D.

F. $x_1 \xrightarrow{E} x_2 = x \cdot a \xrightarrow{E} y \cdot a$ and $x \xrightarrow{E} y \in$ Ax.

F.1. $x_{n+2} \xrightarrow{R} x_{n+3} = (A1)$. Then $x \cdot a \xrightarrow{E} y \cdot a \xrightarrow{O} \cdots \xrightarrow{O} (z/v) \cdot v \xrightarrow{R} z \in$ Ax. We have two possibilities:

F.1.1. $y \xrightarrow{O} \cdots \xrightarrow{O} z/v \in$ Ax and $a \xrightarrow{O} \cdots \xrightarrow{O} v \in$ Ax. If $a \neq v$, we replace our sequence by $x \cdot a \xrightarrow{O} \cdots \xrightarrow{O} x \cdot v \xrightarrow{E} y \cdot v \xrightarrow{O} \cdots \xrightarrow{O} (z/v) \cdot v \xrightarrow{R} z$. If $a = v$, then we must consider all admissible forms of $x \xrightarrow{E} y$:

F.1.1.(a) $x \xrightarrow{E} y = (A2)$. Then instead of $x \cdot a \xrightarrow{E} ((x \cdot t)/t) \cdot a \xrightarrow{O} \cdots \xrightarrow{O} (z/a) \cdot a \xrightarrow{R} z$ we put $x \cdot a \xrightarrow{O} \cdots \xrightarrow{O} x \cdot t \xrightarrow{O} \cdots \xrightarrow{O} z$.

F.1.1.(b) For $x \xrightarrow{E} y$ being (A2′) or (A3) we proceed as in F.1.1.(a).

F.1.1.(c) $x \xrightarrow{E} y$ can not arise by (R2) or (R2′), because we would have $y' \cdot b \xrightarrow{O} \cdots \xrightarrow{O} z/a$ or $b \cdot y' \xrightarrow{O} \cdots \xrightarrow{O} z/a$, which is impossible.

F.1.1.(d) $x \xrightarrow{E} y = x'/b \xrightarrow{E} y'/b$ and $x' \xrightarrow{E} y' \in$ Ax. Then instead of the sequence

$(x'/b) \cdot a \xrightarrow{\text{E}} (y'/b) \cdot a \xrightarrow{\text{O}} \cdot \xrightarrow{\text{O}} (z/a) \cdot a \xrightarrow{\text{R}} z$, in which $y' \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} z$ and $a \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} b$, we put $(x'/b) \cdot a \xrightarrow{\text{E}} (y'/b) \cdot a \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} (y'/b) \cdot b \xrightarrow{\text{R}} y' \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} z$, provided $y' \neq z$, and $(x'/b) \cdot a \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} (x'/b) \cdot b \xrightarrow{\text{R}} y'$, provided $y' = z$.

F.1.1.(e) If $x \xrightarrow{\text{E}} y = b/x' \xrightarrow{\text{E}} b/y'$ and $y' \xrightarrow{\text{R}} x' \in Ax$, then we proceed as in F.1.1.(d).

F.1.2. $y \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} v \in \text{Ax}$ and $a \xrightarrow{\text{O}} \cdots \xrightarrow{\text{O}} z/v \in \text{Ax}$. In this case the proof is similar to that in F.1.1 and is left to the reader.

F.2. $x_{n+2} \xrightarrow{\text{R}} x_{n+3} = (\text{A1}')$. In this case we proceed as in F.1.

F.3. $x_{n+2} \xrightarrow{\text{R}} x_{n+3} = v \cdot u \xrightarrow{\text{R}} z \cdot u$ or $x_{n+2} \xrightarrow{\text{R}} x_{n+3} = u \cdot v \xrightarrow{\text{R}} u \cdot z$ and $v \xrightarrow{\text{R}} z \in \text{Ax}$. These cases are similar to the cases D.1 and D.2 and are left to the reader.

G. If $x_1 \xrightarrow{\text{E}} x_2$ results from (R2′), then we follow the method presented in F. □

It is easily seen that the assumption in Lemma 5 saying that no formula in the sequence may be an instance of (A0) is not essential as the reduced form of any sequence of the index EO^nR can be considered instead of the sequence itself. The point of Lemma 5 is that for a given EO^nR–sequence we can either get rid of the E–formula and the R–formula from this sequence or we can reduce the number of O's between E– and R–formula. Thus Lemma 4 and Lemma 5 give us

Corollary. *Given a derivable sequence* $x_1 \to \cdots \to x_{n+3}$ *of the index* EO^nR *there exists a derivable sequence* $y \to \cdots \to z$ *whose index is either* O^k, *where* $k \leq n+2$, *or* O^mERO^l, *where* $m + l \leq n$, *such that* $x_1 = y$ *and* $x_{n+3} = z$.

Proof. By induction on n. □

If $x \to y$ is a derivable E– (resp. R– or O–) formula, then an instance of the rule (C) employing $x \to y$ is called an E– (resp. R– or O–) *instance* of (C). We call a derivation D of $X \to y$ in NCL *seminormal* if all E–instances of (C) follow R–instances of (C) as well as the rules (A), (A′), (PR) and (PR′). We refer to a derivation D of $X \to y$ in NCL as to *normal* if it is seminormal and additionally if all R–instances of (C) precede (A), (A′), (PR) and (PR′) and no O–rule is placed between (A), (A′), (PR) and (PR′).

Theorem 1. *If* $\vdash_{\text{NCL}} X \to x$, *then any derivation* D of $X \to x$ *can effectively be transformed to a seminormal derivation.*

Proof. We show how to transform to the desired form a derivation in which a sequence of the index EO^n ($n \geq 0$) precedes one of the rules (A), (A′), (PR), (PR′) or an R–instance of (C).

1. A sequence S of the index EO^n ($n \geq 0$) precedes (A):

(a) S precedes the functorial premise of (A). If $n = 0$ we proceed as in the noncommutative case (see [17]). Therefore, we assume that $n > 0$. Then the part of the derivation D of the form

$$
\begin{array}{ll}
\text{(E)} & \dfrac{X \to u/v}{X \to u_1/v_1} \\
\text{(O)} & \overline{\qquad\vdots\qquad} \\
\text{(O)} & \overline{\qquad\qquad} \\
\text{(A)} & \dfrac{X \to u_n/v_n \qquad\qquad Y \to v_n}{(X,Y) \to u_n}
\end{array}
$$

can be transformed to

$$
\begin{array}{lll}
 & \qquad\qquad\qquad\qquad \dfrac{Y \to v_n}{Y \to v_{n-1}} & \text{(O)} \\
 & \qquad\qquad\qquad\qquad \overline{\qquad\vdots\qquad} & \text{(O)} \\
\text{(E)} & \dfrac{X \to u/v}{} \qquad\qquad\quad \overline{\qquad\qquad} & \text{(O)} \\
\text{(A)} & \dfrac{X \to u_1/v_1 \qquad\qquad Y \to v_1}{(X,Y) \to u_1} & \\
\text{(O)} & \dfrac{(X,Y) \to u_1}{(X,Y) \to u_2} & \\
\text{(O)} & \overline{\qquad\vdots\qquad} & \\
\text{(O)} & \dfrac{\qquad\qquad}{(X,Y) \to u_n} &
\end{array}
$$

where $v_n \overset{O}{\to} \cdots \overset{O}{\to} v_1$ and $u_1 \overset{O}{\to} \cdots \overset{O}{\to} u_n$ are reduced forms of the constituents of the sequence S. This transformation reduces the case $n > 0$ to the case $n = 0$.

(b) S precedes the argument premise of (A). If $n = 0$ we proceed as in the noncommutative case. Let $n > 0$. Then instead of the derivation

$$
\begin{array}{lll}
 & \qquad\qquad\qquad\qquad \dfrac{Y \to u_1}{Y \to u_2} & \text{(E)} \\
 & \qquad\qquad\qquad\qquad \overline{\qquad\vdots\qquad} & \text{(O)} \\
 & \qquad\qquad\qquad\qquad \overline{\qquad\qquad} & \text{(O)} \\
\text{(A)} & \dfrac{X \to x/u_n \qquad\qquad Y \to u_n}{(X,Y) \to x} &
\end{array}
$$

we put

$$
\begin{array}{lll}
\text{(E)} & \dfrac{X \to x/u_n}{X \to x/u_{n-1}} & \\
\text{(O)} & \overline{\qquad\vdots\qquad} & \\
\text{(O)} & \overline{\qquad\qquad} \qquad\qquad \dfrac{Y \to u_1}{} & \text{(E)} \\
\text{(A)} & \dfrac{X \to x/u_2 \qquad\qquad Y \to u_2}{(X,Y) \to x} &
\end{array}
$$

Therefore we can treat this reduction as in case $n = 0$.

2. If a sequence of the index EO^n precedes (A′), we can apply the same reasoning as in 1.

3. A sequence S of the index EO^n precedes (PR). It is sufficient to consider only the

case in which S appears before the left premise of (PR) as the argument for the right premise is similar. In the case $n = 0$ we imitate the proof for the noncommutative version (see [17]). Let $n > 0$. Then the fragment of a derivation

$$\begin{array}{ll}
\text{(E)} & \underline{X \to x_1} \\
\text{(O)} & \underline{X \to x_2} \\
 & \quad\vdots \\
\text{(O)} & \underline{\qquad\quad} \\
\text{(PR)} & \dfrac{X \to x_n \qquad\qquad\qquad Y \to y}{(X,Y) \to x_n \cdot y}
\end{array}$$

can be replaced by

$$\begin{array}{ll}
\text{(PR)} & \dfrac{X \to x_1 \qquad\qquad\qquad Y \to y}{\begin{array}{ll}
\text{(E)} & \underline{(X,Y) \to x_1 \cdot y} \\
\text{(O)} & \underline{(X,Y) \to x_2 \cdot y} \\
 & \qquad\vdots \\
\text{(O)} & \underline{\qquad\qquad\qquad} \\
 & (X,Y) \to x_n \cdot y
\end{array}}
\end{array}$$

4. For the rule (PR′) the argument is similar to that of 3.

5. Let a sequence S of the index EO^n precedes an R–instance of the rule (C). If $n = 0$ then we employ Lemma 1 to obtain a sequence of any but ER index. Let $n > 0$. Then, by Corollary , S can be transformed to a sequence of the index O^k, where $k \le n + 2$, or of the index $\mathrm{O}^m\mathrm{ERO}^l$, where $m + l \le n$, with the same initial and terminal types. In the first case we have nothing to do, and in the second case it suffices to apply Lemma 1 as in the case $n = 0$. □

Theorem 2. *If* $\vdash_{\mathrm{NCL}} X \to x$, *then any derivation* D *of* $X \to x$ *can effectively be transformed to a normal form.*

Proof. Let D be a derivation of $X \to x$ in NCL. By Theorem 1 there exists a seminormal derivation D_1 of $X \to x$ in NCL. We have to show that every part of D_1 in which a sequence of the index $\mathrm{O}^n\mathrm{R}$ follows one of the rules (A), (A′), (PR) or (PR′) can be modified in such a way that the R–instance of the rule (C) will change its position according to the definition of normal derivation.

1. Let a sequence of the index $\mathrm{O}^n\mathrm{R}$ follows the rule (A). Then the derivation

$$\begin{array}{ll}
\text{(A)} & \dfrac{X \to x/y \qquad\qquad\qquad Y \to y}{\begin{array}{ll}
\text{(O)} & \underline{(X,Y) \to x} \\
\text{(O)} & \underline{(X,Y) \to x_1} \\
 & \qquad\vdots \\
\text{(O)} & \underline{\qquad\qquad\qquad} \\
\text{(O)} & \underline{(X,Y) \to x_n} \\
 & (X,Y) \to z
\end{array}}
\end{array}$$

in which $x \xrightarrow{\mathrm{O}} x_1 \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} x_n \xrightarrow{\mathrm{R}} z \in \mathrm{Ax}$ can be replaced by

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X \to x/y}{X \to x_1/y}\,(\mathrm{O})}{X \to x_2/y}\,(\mathrm{O})}{\vdots}\,(\mathrm{O})}{X \to x_n/y}\,(\mathrm{O})}{X \to z/y}\,(\mathrm{R}) \qquad Y \to y}{(X,Y) \to z}\,(\mathrm{A})$$

2. The proof for the rule (A′) is similar to that in 1.

3. Assume that a sequence of the index $\mathrm{O}^n\mathrm{R}$ follows the rule (PR). If $n = 0$, then we can change the derivation as in the noncommutative case. If $n > 0$ we must modify the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{X \to x \qquad Y \to y}{(X,Y) \to x \cdot y}\,(\mathrm{PR})}{(X,Y) \to z_1}\,(\mathrm{O})}{\vdots}\,(\mathrm{O})}{(X,Y) \to z_n}\,(\mathrm{O})}{(X,Y) \to z}\,(\mathrm{R})$$

As $x \cdot y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n \in \mathrm{Ax}$ thus z_n must be a product type, say $z_n = z_n^1 \cdot z_n^2$, and according to Lemma 2, either $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n^1 \in \mathrm{Ax}$ and $y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n^2 \in \mathrm{Ax}$ or $x \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n^2 \in \mathrm{Ax}$ and $y \xrightarrow{\mathrm{O}} \cdots \xrightarrow{\mathrm{O}} z_n^1 \in \mathrm{Ax}$. Hence the above derivation can be replaced by

$$\dfrac{\dfrac{\dfrac{\dfrac{X \to x}{\vdots}\,(\mathrm{O})}{X \to z_n^1}\,(\mathrm{O}) \qquad (\mathrm{O})\,\dfrac{\dfrac{Y \to y}{\vdots}\,(\mathrm{O})}{Y \to z_n^2}}{(X,Y) \to z_n^1 \cdot z_n^2}\,(\mathrm{PR})}{(X,Y) \to z}\,(\mathrm{R})$$

or by

$$\dfrac{\dfrac{\dfrac{\dfrac{X \to x}{\vdots}\,(\mathrm{O})}{X \to z_n^2}\,(\mathrm{O}) \qquad (\mathrm{O})\,\dfrac{\dfrac{Y \to y}{\vdots}\,(\mathrm{O})}{Y \to z_n^1}}{(X,Y) \to z_n^1 \cdot z_n^2}\,(\mathrm{PR'})}{(X,Y) \to z}\,(\mathrm{R})$$

In this way we reduce our problem to the case in which an R–instance of (C) is placed directly after (PR) or (PR′). This situation however can be treated as in the noncommutative calculus.

Finally, let us notice that transformations employed in this proof can be used in order to get rid of those O–instances of (C) which are placed between the rules (A), (A′), (PR) or (PR′). □

3 Phrase languages and CF–grammars

Let V be a finite vocabulary. The set BS(V) of *phrase structures over* V is defined as the smallest one such that: (i) $V \subseteq \mathrm{BS}(V)$, (ii) if $A_1, \ldots, A_n \in \mathrm{BS}(V)$, then $(A_1 \ldots A_n) \in \mathrm{BS}(V)$. We will denote by $\mid A \mid$ the sequence arising from a phrase structure A by deleting all brackets. Any subset L of BS(V) will be referred to as a *phrase language over* V.

The set BS(V) provided with operations $f_n(A_1, \ldots A_n) = (A_1 \ldots A_n),\ n = 2, 3, 4, \ldots$ can be considered as an absolutely free algebra over the set of generators V. By a *congruence in* $[\mathrm{BS}(V)]^2$ we mean an equivalence relation $\sim$ preserving operations f_i, i.e. a relation satisfying the condition:

$$\text{if } A_1 \sim B_1, \ldots, A_n \sim B_n, \text{ then } (A_1 \ldots A_n) \sim (B_1 \ldots B_n).$$

If $\sim$ is a congruence in $[\mathrm{BS}(V)]^2$ and $L \subseteq \mathrm{BS}(V)$ then we call $\sim$ a *congruence on* L if and only if for all A, $B \in L$ the following condition is fulfilled:

$$\text{if } A \sim B, \text{ then } (A \in L \leftrightarrow B \in L).$$

The largest with respect to inclusion congruence on a phrase language L, to be denoted INT_L, is called the *intersubstitutability relation for* L. We refer to the index of the relation INT_L as to the *index of* L and denote this number by ind(L).

Given a phrase structure $A = (A_1 \ldots A_n)$ we refer to $A_1, \ldots, A_n$ as to the *direct substructures of* A. The set sub(A) of *substructures of* A is defined in a natural way: (i) $A \in \mathrm{sub}(A)$, (ii) if $B \in \mathrm{sub}(A)$ and C is a direct substructure of A, then $C \in \mathrm{sub}(A)$. The *size of* $A \in \mathrm{BS}(V)$, to be denoted s(A), is the maximum number of direct substructures in any element of sub(A). For $L \subseteq \mathrm{BS}(V)$ we put $\mathrm{s}(L) = \sup\{\mathrm{s}(A) : A \in L\}$ and call s(L) the *size of* L.

By a *path in* $A \in \mathrm{BS}(V)$ we mean a sequence $A_0, A_1, \ldots, A_n$ such that for all $1 \leq i \leq n$, A_i is a direct substructure of A_{i-1}. Structures A_0 and A_n are called the *initial* and *final term of* $A_0, A_1, \ldots, A_n$, respectively, and the number n is referred to as the *length of this path.* The *external degree of* $A \in \mathrm{BS}(V)$, denoted by $\deg^{\mathrm{e}}(A)$ is defined to be the minimal length of paths in A whose initial term is A itself and whose final term is an element from V. We put

$$\deg(A) = \max\{\deg^{\mathrm{e}}(B) : B \in \mathrm{sub}(A)\}$$

and for $L \subseteq \mathrm{BS}(V)$

$$\deg(L) = \sup\{\deg(A) : A \in L\}$$

and call those numbers the *degree of* A and the *degree of* L respectively.

We admit a standard definition of a (λ–free) CF–grammar as an ordered quadruple $\langle V, U, s, P\rangle$ in which symbols V, U, s, P denote, respectively, the set of *terminals*, *nonterminals*, the *initial symbol* and the *set of production rules.* We adopt the notation $a \mapsto b_1 \dots b_n$ for elements of P. A production rule $a \mapsto b_1 \dots b_n$ is called a *permutation variant of* $a \mapsto c_1 \dots c_n$ if for all $i \in \{1, \dots, n\}$ we have $b_i = c_{j_i}$ for some permutation $(j_1, \dots, j_n)$ of $(1, \dots, n)$. In case of a binary rule, i.e. when $n = 2$ we use the term 'transposition variant' instead of 'permutation variant'. A set of production rules is *closed with respect to permutation* if along with a certain rule it also contains all permutation variants of this rule. A CF–grammar (resp. CF–language) is called *closed with respect to permutation* if such is its set of production rules (resp. a CF–grammar generating this language). The definition of a grammar and of a language *closed with respect to transposition* is similar.

Every CF–grammar induces in a natural way a bracketing on elements of the (string) language it generates. This bracketing is fully determined by the production rules of a grammar: each production rule transforms an element of a nonterminal vocabulary into a phrase staying on the right side of the rule. Thus, along with a (string) language $\mathrm{L}(\mathcal{G}) \subseteq V^*$, a CF–grammar $\mathcal{G}$ generates also a *phrase language* $\mathrm{BL}(\mathcal{G}) \subseteq \mathrm{BS}(V)$ and $\mathrm{L}(\mathcal{G}) = \{|\ A\ |\colon A \in \mathrm{BL}(\mathcal{G})\}$. The following theorem provides a necessary and sufficient condition for a phrase language L to be generated by a CF–grammar (see [26]):

Theorem 3. *A phrase language* L *is generated by a* CF*–grammar if and only if* $\mathrm{s}(L)$ *and* $\deg(L)$ *are finite.*

We close the section with an important for our further considerations notion of $\overline{\mathrm{CF}}$–grammar: A CF–grammar $\mathcal{G}$ is called a $\overline{\mathrm{CF}}$*–grammar* if: (i) $\mathcal{G}$ is closed with respect to permutation, (ii) $\deg(\mathrm{BL}(\mathcal{G}))$ is finite.

4 Categorial grammars and their languages

Any calculus of syntactic types, as, for example, one of the described in Section 2, can play the role of a *type reduction system* in a categorial grammar. In what follows we give only most important definitions and results which we use later on, the reader is referred to [10] and [5] as to most comprehensive sources.

A *categorial grammar over a type reduction system* TRS is an ordered quadruple $G = \langle V_G, I_G, s_G, \mathrm{TRS}\rangle$, where V_G is an nonempty *vocabulary of* G, s_G is a *distinguished primitive type* understood as a type of properly built sentences, $I_G \subseteq V_G \times \mathrm{TP}$ is a finite relation called the *initial type assignment of* G. I_G can in a natural way be extended to a relation $F_G \subseteq \mathrm{BS}(V_G) \times$ BSTP. By the *string* (resp. *phrase*) *language* $\mathrm{L}(G)$ (resp. $\mathrm{BL}(G)$) *generated by a categorial grammar* G *over* TRS we mean the set

$$\mathrm{L}(G) = \{|\ A\ |\colon (\exists X \in \mathrm{BSTP})((A, X) \in F_G \ \&\ \vdash_{\mathrm{TRS}} X \to s_G)\}$$

(resp. $\mathrm{BL}(G) = \{ A : (\exists X \in \mathrm{BSTP})((A, X) \in F_G \ \&\ \vdash_{\mathrm{TRS}} X \to s_G)\}$).

As $\mathrm{L}(G) = \{|\ A\ |\colon A \in \mathrm{BL}(G)\}$ thus if two CGs generate the same phrase languages then they also generate the same string languages; the converse implication however does not hold. If there is no special reason we call string languages generated by CGs simply languages. A categorial grammar in which TRS = NA (resp. NCA, NCL) will be referred to as an NA–(resp. NCA–, NCL–)*grammar*. We adopt this convention for product-free calculi as well.

Phrase languages generated by CGs defined as above are of size ≤ 2, this is a consequence of the definition of syntactic calculi where only at most binary phrase structures on left-hand sides of formulas are admitted.

The following theorem establishes the equivalence of NA– and NA^{O}–grammars within the scope of phrase languages, thus within the scope of string languages as well (see [18] where this result is given in a stronger form).

Theorem 4. NA– *and* NA^{O}– *grammars generate the same class of phrase languages.*

For x being a product-free type, let the *order of* x, to be denoted by $\text{o}(x)$, be a non-negative integer defined inductively as follows: (i) $\text{o}(x) = 0$, for $x \in \text{Pr}$, (ii) $\text{o}(x/y) = \text{o}(y\backslash x) = \max\{\text{o}(x), \text{o}(y)+1\}$. By the *order of a product-free categorial grammar* G we mean the number

$$\text{o}(G) = \sup\{\text{o}(x) : (\exists v \in V_G)((v,x) \in I_G)\}$$

The following theorem was proved in [8] (in a stronger case of functorial languages):

Theorem 5. *Any phrase language generated by an* NA^{O}– *grammar is also generated by an* NA^{O}– *grammar* G *such that* $\text{o}(G) \leq 1$.

The next theorem provides a characterization of phrase languages generated by NA^{O}–grammars (see [7] where this result covers a stronger case of functorial languages)

Theorem 6. *A phrase language* L *is generated by an* NA^{O}– *grammar if and only if both* $\text{ind}(L)$ *and* $\deg(L)$ *are finite.*

5 The inclusion of the class of NCL– languages in the class of $\overline{\text{CF}}$– languages

The existence of normal form for derivations in NLC provided by Theorem 2 enables us to construct for every NCL–grammar an NCA–grammar generating the same language. Below we give this construction omitting however the proof of the equality of languages as the aggument is similar to that presented for noncommutative calculi, see [7] and [17].

Let $G_1 = \langle V_{G_1}, I_{G_1}, s_{G_1}, \text{NCL}\rangle$ be an NCL–grammar. We construct an NCA–grammar $G_2 = \langle V_{G_2}, I_{G_2}, s_{G_2}, \text{NCA}\rangle$ as follows: $V_{G_1} = V_{G_2}, s_{G_1} = s_{G_2}$, and for $x \in \text{TP}$ and $v \in V_{G_2}$ let $(v,x) \in I_{G_2}$ if and only if there exist types $x_1, \ldots, x_n$ such that $(v, x_1) \in I_{G_1}, x_n = x$, $x_1 \to \ldots \to x_n \in \text{Ax}$ and every formula in this sequence is either R– or O–formula.

Lemma 6. $L(G_1) = L(G_2)$.

Having reduced NCL–grammars to NCA–grammars in the way which preserves generated languages we can make the next step: construct a CF–grammar which generates the same (phrase) language as a given NCA–grammar. The details of this construction are as follows: For an NCA–grammar $G = \langle V_G, I_G, s_G, \text{NCA}\rangle$ we define a CF–grammar $\mathcal{G}$ in the following way: $V_{\mathcal{G}} = V_G, s_{\mathcal{G}} = s_G$ and $U_{\mathcal{G}} = \text{sub}(\{x \in \text{TP}: (\exists v \in V_G)((v,x) \in I_G\})$. The set $P_{\mathcal{G}}$ of production rules contains all rules of one of the shapes: (i) $x \mapsto v$ where $x \in \text{TP}$, $v \in V_G$, and $(v,x) \in I_G$, or (ii) $x \mapsto x/y\ y$, $x \mapsto y\ x/y$, $x \cdot y \mapsto x\ y$, $x \cdot y \mapsto y\ x$, for all $x, y \in U_{\mathcal{G}}$. We have

Lemma 7. $\text{BL}(\mathcal{G}) = \text{BL}(G)$.

The proof of this lemma is essentially the same as in the noncommutative calculus NA, see [17].

The following lemma is a straightforward consequence of Lemma 7 and the presented construction:

Lemma 8. *For every* NCA–*grammar* G *one finds a closed with respect to transposition* CF–*grammar* $\mathcal{G}$ such that $\text{BL}(G) = \text{BL}(\mathcal{G})$.

Lemma 9. *Let* G *be an* NCA–*grammar. If* $\mathcal{G}$ *is a* CF–*grammar constructed as above, then* $\deg(\text{BL}(\mathcal{G})) < \aleph_0$

Proof. By Lemma 7, $\text{BL}(\mathcal{G}) = \text{BL}(G)$, thus it is sufficient to show the finiteness of $\text{BL}(G)$. Let $G = \langle V_G, I_G, s_G, \text{NCA}\rangle$ and let $\overline{\text{NCA}}$ denotes the calculus obtained from NCA by dropping the rules (A′) and (PR′), thus employing only (A) and (PR). We put $\overline{G} = \langle V_G, I_G, s_G, \overline{\text{NCA}}\rangle$ and immediately get $\text{BL}(\overline{G}) \subseteq \text{BL}(G)$. Observe that $\text{BL}(G)$ arises from $\text{BL}(\overline{G})$ by adding all phrase structures obtained by transpositions of direct substructures in substructures of elements of $\text{BL}(\overline{G})$. However, the degree of any phrase structure B obtained by transpositions from a given structure A is the same as the degree of the structure A itself. Thus $\deg(\text{BL}(\overline{G})) = \deg(\text{BL}(G))$ and we will show that $\deg(\text{BL}(\overline{G}))$ is finite. In order to do it we prove that for some phrase language L_0 such that $\text{BL}(\overline{G}) \subseteq L_0$ we have $\deg(L_0) < \aleph_0$. We put the calculus NA instead of $\overline{\text{NCA}}$ in the definition of $\overline{G}$ and denote the obtained grammar by G_0. Let $L_0 = \text{BL}(G_0)$. Every formula derivable in $\overline{\text{NCA}}$ is also derivable in NA, thus we get the inclusion $\text{BL}(\overline{G}) \subseteq L_0$. By Theorem 4, the language L_0 being generated by an NA–grammar is also generated by an NA^{O}–grammar. Consequently, according to Theorem 6, the number $\deg(L_0)$ is finite. Thus we have $\deg(\text{BL}(\mathcal{G})) = \deg(\text{BL}(G)) = \deg(\text{BL}(\overline{G})) \leq \deg(L_0) < \aleph_0$. □

Lemmas 6,7 and 9 give

Theorem 7. *For every* NCL–*grammar* G *one finds an* $\overline{\text{CF}}$–*grammar* $\mathcal{G}$ *such that* $\text{L}(G) = \text{L}(\mathcal{G})$, *i.e. the class of languages generated by* NCL–*grammars is included in the class of languages generated by* $\overline{\text{CF}}$–*grammars.*

6 The equivalence of $\overline{\mathrm{CF}}$–grammars and NLC–grammars.

Lemma 10. *For every* CF*– grammar* $\mathcal{G}$ *one can find a* CF*– grammar* $\mathcal{G}'$ *such that* $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(\mathcal{G}')$ *and all the rules of production in* $\mathcal{G}'$ *are of one of the forms:* $a \mapsto b_1 \ldots b_n$ *or* $a \mapsto v$*, where* $a, b_1, \ldots, b_n$ *are nonterminals and* v *is a terminal in* $\mathcal{G}'$*.*

Proof. This lemma usually constitutes a part of the proof of the Chomsky normal form theorem, see for example [15]. The equality $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(\mathcal{G}')$ is a consequence of the fact that the employed in the proof procedures of getting rid of unit productions (i.e. of productions of the form $a \mapsto b$) as well as of productions containing terminals on right-hand sides do not affect the phrase structure of elements of the generated language. □

Lemma 11. *Let* $\mathcal{G} = \langle V_\mathcal{G}, U_\mathcal{G}, s_\mathcal{G}, P_\mathcal{G}\rangle$ *be a* $\overline{\mathrm{CF}}$*– grammar. There exist then a closed with respect to transposition* CF*– grammar* $\mathcal{G}'$ *in Chomsky normal form such that* $\mathrm{L}(\mathcal{G}) = \mathrm{L}(\mathcal{G}')$ *and* $\deg(\mathrm{BL}(\mathcal{G}') < \aleph_0$.

Proof. According to Lemma 10 we can assume that $P_\mathcal{G} = \overline{\overline{P_\mathcal{G}}} \cup \overline{P_\mathcal{G}}$, where $\overline{\overline{P_\mathcal{G}}}$ consists of productions of the form $a \mapsto b_1 \ldots b_n, n \geq 2$ and $\overline{P_\mathcal{G}}$ consists of productions of the form $a \mapsto v$, where $a, b_1, \ldots, b_n \in U_\mathcal{G}$ and $v \in V_\mathcal{G}$. We show that every set of production rules which comprises all permutation variants of a given production rule can be replaced by a closed with respect to transposition set of binary rules. Let $R = a \mapsto b_1 \ldots b_n \in \overline{\overline{P_\mathcal{G}}}$. If $n = 2$, then there is nothing to show as, according to our assumptions, both $a \mapsto b_1 b_2$ and $a \mapsto b_2 b_1$ are in $\overline{\overline{P_\mathcal{G}}}$ and they constitute the desired set of binary rules. For $n \geq 3$ we replace R by a set F_R^0 consisting of the rules $a \mapsto b_1 c_1,\ c_1 \mapsto b_2 c_2, \ldots, c_{n-2} \mapsto b_{n-1} b_n$ as we usually do in the construction of the Chomsky normal form for CF– grammar ($c_1, \ldots, c_{n-2}$ are new nonterminals). Then we add to F_R^0 all transposition variants of its elements and denote the set obtained in this way by F_R. Observe that for generation of a language over $V_\mathcal{G}$ only those strings derivable from a are essential which consist of nonterminals $b_1, \ldots, b_n$ but not $c_1, \ldots, c_{n-2}$. However, due to the form of the rules in F_R, every string derivable from a by means of those rules which consist exclusively of nonterminals $b_1, \ldots, b_n$ must contain all of them, additionally, along with a sequence $b_1 \ldots b_n$, some of its permutations can also be derived from a by means of productions from F_R. Thus, F_R is a substitute for the rule R as well as for some of its permutation variants. The described procedure can be performed for all permutation variants of R (all new nonterminals must differ one from another in order to avoid an interaction of rules). The set of all binary rules obtained in this way for R and all its permutation variants will be denoted by $F_{Perm(R)}$. This set produces the same strings as the rule R and its permutation variants and no other strings. $F_{Perm(R)}$ is also closed with respect to transposition. We define $\overline{P_{\mathcal{G}'}} = \overline{P_\mathcal{G}}$, $\overline{\overline{P_{\mathcal{G}'}}} = \bigcup\{F_{Perm(R)} : R \in \overline{\overline{P_\mathcal{G}}}\}$, $P_{\mathcal{G}'} = \overline{\overline{P_{\mathcal{G}'}}} \cup \overline{P_{\mathcal{G}'}}$ $s_{\mathcal{G}'} = s_\mathcal{G}$, $V_{\mathcal{G}'} = V_\mathcal{G}$, and let $U_{\mathcal{G}'}$ consists of all nonterminals from $U_\mathcal{G}$ as well as of all new nonterminals introduced in the process of constructing the sets $F_{Perm(R)}$ for all R's. We put $\mathcal{G}' = \langle V_{\mathcal{G}'}, U_{\mathcal{G}'}, s_{\mathcal{G}'}, P_{\mathcal{G}'}\rangle$. The set $\overline{\overline{P_{\mathcal{G}'}}}$ produces precisely the same strings over $U_{\mathcal{G}'}$ as $\overline{\overline{P_\mathcal{G}}}$ over $U_\mathcal{G}$ and consequently, as $\overline{P_{\mathcal{G}'}} = \overline{P_\mathcal{G}}$, we have $\mathrm{L}(\mathcal{G}') = \mathrm{L}(\mathcal{G})$.

Now we show that $\deg(\mathrm{BL}(\mathcal{G}')) < \aleph_0$. The replacement of a rule $a \mapsto b_1 \dots b_n$, $n \geq 3$ by a set of binary rules introduces a (binary) phrase structure on $b_1 \dots b_n$ and consequently makes the phrase structure of elements of $\mathrm{BL}(\mathcal{G}')$ finer than that we have in $\mathrm{BL}(\mathcal{G})$. As a result, the length of paths leading from any substructure of an element of $\mathrm{BL}(\mathcal{G}')$ to an atom (terminal) can increase. However, for any $A \in \mathrm{BL}(\mathcal{G})$ and $A' \in \mathrm{BL}(\mathcal{G}')$ such that $\mid A \mid = \mid A' \mid$ we have $\deg(A') \leq \deg(A)\cdot(\mathrm{s}(\mathrm{BL}(G)) - 1) < \deg(A)\cdot \mathrm{s}(\mathrm{BL}(\mathcal{G}))$. But $\mathrm{s}(\mathrm{BL}(\mathcal{G}))$ is finite (it is the maximal length of strings on the right-hand sides of production rules from $P_{\mathcal{G}}$) and $\deg(\mathrm{BL}(\mathcal{G}))$ is finite as well ($\mathcal{G}$ is a $\overline{\mathrm{CF}}$-grammar). Therefore $\deg(\mathrm{BL}(\mathcal{G})) = \sup\{\deg(A') : A' \in \mathrm{BL}(\mathcal{G}')\} \leq \sup\{\deg(A)\cdot\mathrm{s}(\mathrm{BL}(\mathcal{G})) : A \in \mathrm{BL}(\mathcal{G})\ \&\ \mid A \mid = \mid A' \mid\} = \mathrm{s}(\mathrm{BL}(\mathcal{G})\cdot\sup\{\deg(A) : A \in \mathrm{BL}(\mathcal{G})\} = \deg(\mathrm{BL}(\mathcal{G}))\cdot\mathrm{s}(\mathrm{BL}(\mathcal{G})) < \aleph_0$. □

Lemma 12. *If $\mathcal{G}$ is a $\overline{\mathrm{CF}}$-grammar in Chomsky normal form, then* $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(G)$ *for some* $\mathrm{NCA}^{\mathrm{O}}$*-grammar G of order* ≤ 1.

Proof. By Theorem 3, $\mathrm{ind}(\mathrm{BL}(\mathcal{G})) < \aleph_0$ and $\mathrm{s}(\mathrm{BL}(\mathcal{G})) < \aleph_0$ (the second inequality is not important because in our case we have $\mathrm{s}(\mathrm{BL}(\mathcal{G})) \leq 2$). Since $\deg(\mathrm{BL}(\mathcal{G})) < \aleph_0$, by Theorem 6 we conclude that $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(G_0)$, for some NA^{O}-grammar $G_0 = \langle V_{G_0}, I_{G_0}, s_{G_0}, \mathrm{NA}^{\mathrm{O}}\rangle$. According to Theorem 5 we can assume that $\mathrm{o}(G_0) \leq 1$. If we add to the rules of NA^{O} the rule of permutation (Perm), then we get the product-free version $\mathrm{NCA}^{\mathrm{O}}$ of NCA. Let $G = \langle V_{G_0}, I_{G_0}, s_{G_0}, \mathrm{NCA}^{\mathrm{O}}\rangle$. We have $\mathrm{o}(G) \leq 1$ and, as $\mathrm{NCA}^{\mathrm{O}}$ is stronger than NA^{O}, $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(G_0) \subseteq \mathrm{BL}(G)$. The converse inclusion follows from the fact that $\mathcal{G}$ is closed with respect to transposition, consequently the language $\mathrm{BL}(\mathcal{G})$ (and $\mathrm{BL}(G_0)$) is closed with respect to transpositions of its substructures. Thus adding the rule (Perm) to NA^{O} i.e. employing $\mathrm{NCA}^{\mathrm{O}}$ as a type reduction system instead of NA^{O} will not lead us beyond the language generated by G_0. □

Lemma 13. *For every* $\mathrm{NCA}^{\mathrm{O}}$*-grammar G_0 of the order* ≤ 1 *there exists an* NCL*-grammar G_1 such that* $\mathrm{BL}(G) = \mathrm{BL}(G_1)$.

Proof. We adopt a standard argument presented for example in [7] or in [18] but suited here to the case of commutative calculi. The axiomatization of $\mathrm{NCA}^{\mathrm{O}}$ we use consists of the axiom scheme (A0) and the rules (A) and (A′). Given an $\mathrm{NCA}^{\mathrm{O}}$-grammar $G = \langle V_G, I_G, s_G, \mathrm{NCA}^{\mathrm{O}}\rangle$ we put $G_1 = \langle V_G, I_G, s_G, \mathrm{NCL}\rangle$ and claim that $\mathrm{BL}(G) = \mathrm{BL}(G_1)$. In order to obtain this equality it is sufficient to prove that $\vdash_{\mathrm{NCA}^{\mathrm{O}}} X \to s_G$ if and only if $\vdash_{\mathrm{NCL}} X \to s_G$ unless all types on X are product-free and of order ≤ 1. It is obvious that every formula derivable in $\mathrm{NCA}^{\mathrm{O}}$ is also derivable in NCL because $\mathrm{NCA}^{\mathrm{O}}$ is a subsystem of NCL. To prove the converse implication let us assume $\vdash_{\mathrm{NCL}} X \to s_G$. By Theorem 2 the formula $X \to s_G$ possesses a normal derivation D in NCL. Since $s_G \in \mathrm{Pr}$, no E–instances of (C)-rule occur in D. For every type x in X any R–instance of (C)-rule would employ such a formula $x \to y$ from Ax that $c(x) > c(y)$. This formula can not be obtained by means of the rules (R2) or (R2′) because of the presence of the product sign $\cdot$ in their conclusions. For axioms (A1), (A1′), (A2), (A2′) as well as for formulas in Ax which arise from them by an application of rules (R1) or (R1′), one sees that this side of a formula which is of greater complexity contains also a product sign. The formula $x \to y$ can not be obtained from (A4) by any rule from Ax as well, otherwise we would have $c(x) = c(y)$. The only remaining possibility of constructing $x \to y$ is that using (A3) and the rules (R1) or (R1′). For any $z, t \in \mathrm{Tp}$ we have however $\mathrm{o}(z/(z/t)) = \max(\mathrm{o}(z), \mathrm{o}(t)+2)$, and thus $\mathrm{o}(z/(z/t)) \geq 2$. Consequently, this side of the formula $x \rightarrow y$ which is of greater complexity would have the order ≥ 2, but for $x \rightarrow y$ being an R–formula this is impossible as $\mathrm{o}(G) \leq 1$. We conclude that no R–formulas are employed in D, thus D is a derivation in NCA. But the rules (PR) and (PR′) can not be applied in D, otherwise for some types z, t a product type $z \cdot y$ would be a subtype of a type in X. As a result D is a derivation in $\mathrm{NCA}^{\mathrm{O}}$ and $\vdash_{\mathrm{NCA}^{\mathrm{O}}} X \rightarrow s_G$. □

Theorem 8. *For any* $\overline{\mathrm{CF}}$*– grammar* $\mathcal{G}$ *there exists an* NCL*– grammar* G *such that* $\mathrm{L}(G) = \mathrm{L}(\mathcal{G})$.

Proof. We conclude from Lemma 10 and Lemma 11 that for the grammar $\mathcal{G}$ one can construct a CF– grammar $\mathcal{G}'$ in Chomsky normal form, closed with respect to transposition and such that $\mathrm{L}(\mathcal{G}) = \mathrm{L}(\mathcal{G}')$ and $\deg(\mathrm{BL}(\mathcal{G}')) < \aleph_0$. Employing Lemma 12 we find for $\mathcal{G}'$ an $\mathrm{NCA}^{\mathrm{O}}$– grammar G_1 such that $\mathrm{o}(G_1) \leq 1$ and $\mathrm{BL}(\mathcal{G}') = \mathrm{BL}(G_1)$. By Lemma 13 we can find for G_1 an NCL– grammar G such that $\mathrm{BL}(G_1) = \mathrm{BL}(G)$. Accordingly, as $\mathrm{L}(\mathcal{G}') = \mathrm{L}(G_1) = \mathrm{L}(G)$, G is the grammar fulfilling the thesis. □

Theorem 9. NLC*– grammars and* $\overline{\mathrm{CF}}$*– grammars generate the same class of (string) languages.*

Proof. This is a consequence of Theorem 7 and Theorem 9. □

Note. It is not known however, whether the finiteness of the degree in the definition of $\overline{\mathrm{CF}}$– grammars is an essential restriction. So far we do not know if the question whether for every CF– grammar $\mathcal{G}$ one can construct an CF– grammar $\mathcal{G}'$ such that $\mathrm{BL}(\mathcal{G}) = \mathrm{BL}(\mathcal{G}')$ and $\deg(\mathrm{BL}(\mathcal{G}')) < \aleph_0$ has a positive or negative answer.

References

[1] Bach, E. (1984): 'Some generalizations of categorial grammars.' In [21], 1-23.

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

[3] Benthem, J. van (1988): 'The Lambek calculus.' In [24], 35-68.

[4] Benthem, J. van (1989): 'Semantic type change and syntactic recognition.' In [11], 231-249.

[5] Benthem, J. van (1991): *Language in Action. Categories, Lambdas and Dynamic Logic,* Studies in Logic and the Foundations of Mathematics, North–Holland, Amsterdam.

[6] Buszkowski, W. (1985): 'The equivalence of unidirectional Lambek categorial grammars and context-free grammars.' *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **31**, 369-384.

[7] Buszkowski, W. (1986): 'Generative capacity of nonassociative Lambek calculus.' *Bulletin of the Polish Academy of Sciences: Mathematics*, **34**, 507-516.

[8] Buszkowski, W. (1986): 'Typed functorial languages.' *Bulletin of the Polish Academy of Sciences: Mathematics*, **34**, 495-505.

[9] Buszkowski, W. (1991): 'On generative capacity of the Lambek calculus.' In [13], 139-152.

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

[11] Chierchia, G., B. Partee and R. Turner (eds.) (1989): *Properties, Types and Meaning*, vol I: *Foundational Issues*, vol II: *Semantic Issues*, Kluwer, Dordrecht.

[12] Došen, K. (1990): 'Modal logic as metalogic.' SNS Report, Tübingen.

[13] Eijck, J. van (ed.) (1991): *Logics in* AI, Lecture Notes in Artificial Intelligence, Springer Verlag, Berlin–Heidelberg–New York.

[14] Gentzen, G. (1934–1935): 'Untersuchungen über das logische Schliessen I–II.' *Mathematische Zeitschrift*, **39**, 176-210, 405-431.

[15] Hopcroft, J. and J. Ullman (1979): *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Company, Reading, Massachusetts.

[16] Jacobson, R. (ed.) (1961): *Structure of Language and Its Mathematical Aspects*, Amer. Math. Soc., Providence, R.I.

[17] Kandulski, M. (1988): 'The equivalence of nonassociative Lambek categorial grammars and context-free grammars.' *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **34**, 41-52.

[18] Kandulski, M. (1988): 'Phrase structure languages generated by categorial grammars with product.' *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, **34**, 373-383.

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

[20] Lambek, J. (1961): 'On the calculus of syntactic types.' In [16], 166-178.

[21] Landman, F. and F. Veltman (1984): *Varieties of Formal Semantics*, Foris, Dordrecht.

[22] Moortgat, M. (1992): 'Labelled Deductive Systems for categorial theorem proving.' OTS Working Papers, OTS-WP-CL-92-003, Research Institute for Language and Speech, Rijksuniversiteit Utrecht.

[23] Moortgat, M. and G. Morrill (1991): 'Heads and phrases. Type calculus for dependency and constituent structure.' ms. OTS, Utrecht, (to appear in *Journal of Logic, Language and Information*).

[24] Oehrle, R., E. Bach and D. Wheeler (eds.) (1988): *Categorial Grammars and Natural Language Structures*, Studies in Linguistics and Philosophy, D. Reidel, Dordrecht.

[25] Pentus, M., (1992): 'Lambek grammars are context–free.' Unpublished manuscript.

[26] Thatcher, J.W. (1967): 'Characterizing derivation trees of context-free grammars through a generalization of finite automata theory.' *Journal Comput. Systems Sci.*, **1**, 317-322.

[27] Troelstra, A.S. (1990): *Lectures on Linear Logic*, ITLI Prepublications X-90-15, University of Amsterdam.

[28] Zielonka, W. (1978): 'A direct proof of the equivalence of free categorial grammars and simple phrase structure grammars.' *Studia Logica*, **37**, 41-57.

The ILLC Prepublication Series

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
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

1992 — Annual Report 1991

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

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