Institute for Language, Logic and Information

THE LAMBEK CALCULUS ENRICHED WITH ADDITIONAL CONNECTIVES

Makoto Kanazawa

ITLI Prepublication Series
for Logic, Semantics and Philosophy of Language LP-91-04

University of Amsterdam

The ITLI Prepublication Series

1986
86-01 The Institute of Language, Logic and Information
86-02 Peter van Emde Boas A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem Logical Syntax

1987
87-01 Jeroen Groenendijk, Martin Stokhof Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem Polyadic quantifiers
87-05 Víctor Sánchez Valencia Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem Categorial Grammar and Type Theory
87-08 Renate Bartsch The Construction of Properties under Perspectives
87-09 Herman Hendriks Type Change in Semantics: The Scope of Quantification and Coordination

1988
Logic, Semantics and Philosophy of Language:
LP-88-01 Michiel van Lambalgen Algorithmic Information Theory
LP-88-02 Yde Venema Expressiveness and Completeness of an Interval Tense Logic
LP-88-03 Year Report 1987
LP-88-04 Reinhard Muskens Going partial in Montague Grammar
LP-88-05 Johan van Benthem Logical Constants across Varying Types
LP-88-06 Johan van Benthem Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe A Blissymbolics Translation Program

Mathematical Logic and Foundations:
ML-88-01 Jaap van Oosten Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:
CT-88-01 Ming Li, Paul M.B.Vitanyi Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks, Gerard R. Renardel de Lavalette Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars, Leen Torenvliet, Peter van Emde Boas Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet, Peter van Emde Boas Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel, Peter van Emde Boas Towards implementing RL

Other prepublications:
X-88-01 Marc Jumelet On Solovay's Completeness Theorem

1989
Logic, Semantics and Philosophy of Language:
LP-89-01 Johan van Benthem The Fine-Structure of Categorial Semantics
LP-89-02 Jeroen Groenendijk, Martin Stokhof Dynamic Predicate Logic, towards a compositional, non-representational semantics of discourse
LP-89-03 Yde Venema Two-dimensional Modal Logics for Relation Algebras and Temporal Logic of Intervals
LP-89-04 Johan van Benthem Language in Action
LP-89-05 Johan van Benthem Modal Logic as a Theory of Information
LP-89-06 Andreja Prijatelj Intensional Lambek Calculi: Theory and Application
LP-89-07 Heinrich Wansing The Adequacy Problem for Sequential Propositional Logic
LP-89-08 Víctor Sánchez Valencia Peirce's Propositional Logic: From Algebra to Graphs
LP-89-09 Zhisheng Huang Dependency of Belief in Distributed Systems

Mathematical Logic and Foundations:
ML-89-01 Dick de Jongh, Albert Visser Explicit Fixed Points for Interpretability Logic
ML-89-02 Roel de Vrijer Extending the Lambda Calculus with Surjective Pairing is conservative
ML-89-03 Dick de Jongh, Franco Montagna Rosser Orderings and Free Variables
ML-89-04 Dick de Jongh, Marc Jumelet, Franco Montagna On the Proof of Solovay's Theorem
ML-89-05 Rineke Verbrugge Σ-completeness and Bounded Arithmetic
ML-89-06 Michiel van Lambalgen The Axiomatization of Randomness
ML-89-07 Dirk Roorda Elementary Inductive Definitions in HA: from Strictly Positive towards Monotone
ML-89-08 Dirk Roorda Investigations into Classical Linear Logic
ML-89-09 Alessandra Carbone Provable Fixed points in $I\Delta_0+\Omega_1$

Computation and Complexity Theory:
CT-89-01 Michiel H.M. Smid Dynamic Deferred Data Structures
CT-89-02 Peter van Emde Boas Machine Models and Simulations
CT-89-03 Ming Li, Herman Neuféglise, Leen Torenvliet, Peter van Emde Boas On Space Efficient Simulations
CT-89-04 Harry Buhrman, Leen Torenvliet A Comparison of Reductions on Nondeterministic Space
CT-89-05 Pieter H. Hartel, Michiel H.M. Smid, Leen Torenvliet, Willem G. Vree A Parallel Functional Implementation of Range Queries
CT-89-06 H.W. Lenstra, Jr. Finding Isomorphisms between Finite Fields
CT-89-07 Ming Li, Paul M.B. Vitanyi A Theory of Learning Simple Concepts under Simple Distributions and Average Case Complexity for the Universal Distribution (Prel. Version)
CT-89-08 Harry Buhrman, Steven Homer, Leen Torenvliet Honest Reductions, Completeness and Nondeterminstic Complexity Classes
CT-89-09 Harry Buhrman, Edith Spaan, Leen Torenvliet On Adaptive Resource Bounded Computations
CT-89-10 Sieger van Denneheuvel The Rule Language RL/1
CT-89-11 Zhisheng Huang, Sieger van Denneheuvel, Peter van Emde Boas Towards Functional Classification of Recursive Query Processing

Other Prepublications:
X-89-01 Marianne Kalsbeek An Orey Sentence for Predicative Arithmetic
X-89-02 G. Wagemakers New Foundations: a Survey of Quine's Set Theory
X-89-03 A.S. Troelstra Index of the Heyting Nachlass
X-89-04 Jeroen Groenendijk, Martin Stokhof Dynamic Montague Grammar, a first sketch
X-89-05 Maarten de Rijke The Modal Theory of Inequality
X-89-06 Peter van Emde Boas Een Relationele Semantiek voor Conceptueel Modelleren: Het RL-project

1990 SEE INSIDE BACK COVER

Instituut voor Taal, Logica en Informatie
Institute for Language, Logic and Information

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Plantage Muidergracht 24
1018TV Amsterdam

Faculteit der Wijsbegeerte
(Department of Philosophy)
Nieuwe Doelenstraat 15
1012CP Amsterdam

THE LAMBEK CALCULUS ENRICHED WITH ADDITIONAL CONNECTIVES

Makoto Kanazawa
Department of Linguistics
Stanford University

ITLI Prepublications
for Logic, Semantics and Philosophy of Language
ISSN 0924-2082

Received August 1991

The Lambek Calculus Enriched with Additional Connectives*

Makoto Kanazawa
Department of Linguistics
Stanford University
Stanford, CA 94305–2150
kanazawa@csli.stanford.edu

August 9, 1991

The Lambek calculus [10, 2], which underlies a flexible version of categorial grammar, is a special kind of implicational logic, with its slashes (/, \) corresponding to logical implication. Viewed from this perspective, it makes sense to add other connectives to it, for example, those corresponding to conjunction and disjunction in more standard logics. The present paper investigates some of the formal properties of this enrichment, especially with respect to the recognizing power of the resulting extended categorial grammars.

§1. Preliminaries

1.1. The Lambek calculus: the logic underlying flexible categorial grammar

A prominent feature of the recent revival of categorial grammar is its 'flexible' character. In addition to the *functional application* of the classical categorial grammar of Ajdukiewicz and Bar-Hillel, various other modes of type combination are employed. An example of an additional mode of combination is *functional composition*:

$$c/b,\ b/a \ \Rightarrow\ c/a, \qquad a\backslash b,\ b\backslash c \ \Rightarrow\ a\backslash c.$$

A different way of looking at this is in terms of *type change*, whereby c/b and $b\backslash c$ change to $(c/a)/(b/a)$ and $(a\backslash b)\backslash(a\backslash c)$, respectively. Another example of type change scheme frequently employed is *type raising*:

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

which allows the functor and the argument to switch roles.

As early as back in 1958, Lambek [10] proposed a certain deductive system, called the Lambek calculus, which derives all of the above schemes (and more) as theorems. A perspicuous presentation of the (product-free) Lambek calculus $\mathsf{L}(/,\backslash)$ can be given in the form of a sequent calculus. Its formulas are atomic ones plus those built up from them using / and \. Formulas are also called types. Expressions of the form $a_1, \ldots, a_n \Rightarrow b$, where each a_i and b are formulas, are *sequents*. Their intuitive meaning is: types $a_1, \ldots, a_n$ combine

*I would like to thank Professor Johan van Benthem for much encouragement and a number of helpful suggestions. Thanks are also due to Dirk Roorda, who commented on an earlier draft of the paper, and to my advisor Professor Stanley Peters.

in this order to yield type b. A sequent is derivable if it can be obtained from axiomatic sequents by (repeated) applications of rules of inference.[1]

Axioms: $a \Rightarrow a$

Rules of Inference:

$$(/\Rightarrow)\ \frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, b/a, X, Z \Rightarrow c} \qquad (\Rightarrow /)\ \frac{X, a \Rightarrow b}{X \Rightarrow b/a}$$

$$(\backslash\Rightarrow)\ \frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, X, a\backslash b, Z \Rightarrow c} \qquad (\Rightarrow \backslash)\ \frac{a, X \Rightarrow b}{X \Rightarrow a\backslash b}$$

$$\textit{Cut:}\ \frac{X \Rightarrow a \quad Y, a, Z \Rightarrow b}{Y, X, Z \Rightarrow b}$$

In $(\Rightarrow /)$ and $(\Rightarrow \backslash)$, X must not be empty, so that all derivable sequents will have non-empty antecedents. There are no structural rules other than Cut. Moreover, Cut is eliminable, which can be proved by means of the standard technique [10].

The idea of the Lambek calculus is based on a certain natural semantics in terms of sets of expressions [10, 11, 2]. A *language model* L assings to each basic type a a set L_a of expressions over a fixed finite alphabet. (Members of L_a are expressions of type a.) On the basis of this, sets of expressions are assigned to complex types of the form b/a or $a\backslash b$ in the following way:

$$L_{b/a} = L_b/L_a = \{y \mid \forall x \in L_a\ yx \in L_b\},$$
$$L_{a\backslash b} = L_a\backslash L_b = \{y \mid \forall x \in L_a\ xy \in L_b\}.$$

(That is, y is an expression of type b/a if for all expressions x of type a, yx is an expression of type b.) A sequent $a_1, \ldots, a_n \Rightarrow b$ is valid iff for all language models L, if x_i is an expression in L_{a_i} $(1 \leq i \leq n)$, then $x_1 \ldots x_n$ is in L_b (i.e., $L_{a_1} \bullet \cdots \bullet L_{a_n} \subseteq L_b$, where $\bullet$ stands for concatenation). Buszkowski [3] proves the completeness of the product-free Lambek calculus with respect to language models. Thus, the Lambek calculus provides a natural metatheory for categorial grammars.

1.2. Categorial grammars

A Lambek categorial grammar ($\mathsf{L}(/,\backslash)$-grammar) consists of an assignment of a finite set of types to each symbol in the alphabet, and one distinguished type (analogous to the start symbol in context-free grammars), where types are formulas of $\mathsf{L}(/,\backslash)$. A $\mathsf{L}(/,\backslash)$-grammar G recognizes a string of symbols $s_1 \ldots s_n$ if there is a matching sequence of types $a_1, \ldots, a_n$ that derives the distinguished type t of G in $\mathsf{L}(/,\backslash)$ (i.e., $a_1, \ldots, a_n \Rightarrow t$ is a derivable sequent in $\mathsf{L}(/,\backslash)$), such that each a_i is a type assigned to s_i by G. A $\mathsf{L}(/,\backslash)$-grammar G recognizes a language L if L is the set of strings that G recognizes ($L(G)$). We say that $\mathsf{L}(/,\backslash)$ recognizes a language L if there is a $\mathsf{L}(/,\backslash)$-grammar that recognizes L.

It is known that $\mathsf{L}(/,\backslash)$ recognizes all context-free languages.[2] The question whether the converse holds—whether all $\mathsf{L}(/,\backslash)$-recognizable languages are context-free—has not been settled so far, although it has been conjectured that it does, and Buszkowski [4, 5, 6] has

[1] In what follows, we assume familiarity with the basic methods and terminology of Gentzen-style proof theory.

[2] Throughout the paper, we use 'context-free languages' to mean context-free languages without ϵ (the empty string).

obtained some partial results. Therefore, the exact recognizing power of $\mathsf{L}(/,\backslash)$—exactly where it lies in the Chomsky hierarchy—remains an open question.

It is sometimes convenient to think of languages over an alphabet consisting of types. Define $L(T,b)$ to be $\{X \in T^* \mid X \Rightarrow b \text{ is derivable}\}$. Then $L(G)$ is obtained from $L(T,t)$ by substitution, where T is the set of types assigned to some symbol by G and t is the distinguished type of G. Properties of the 'type language' $L(T,t)$ may be transferred to the 'symbol language' $L(G)$. For example, if $L(T,t)$ is context-free, so is $L(G)$.

1.3. Variation of the underlying logic

Its naturalness notwithstanding, the original Lambek calculus is by no means the only calculus that can sensibly serve as the underlying calculus of categorial grammars. From a logical point of view, extensions and modifications of the Lambek calculus are worth considering, especially because the Lambek calculus is an extremely impoverished system, as compared to more standard logics. By varying the underlying calculus, we obtain a generalized conception of categorial grammars. All notions defined in the previous section allow straightforward generalization in this respect. For example, calculus K recognizes language L if there is a K-grammar G that recognizes L, where the derivability mentioned in the definition of recognition of strings is derivability in K.

One striking feature of the Lambek calculus is its complete lack of structural rules (except for Cut, which is eliminable). Standard sequent systems employ structural rules like the following:

$$\mathsf{P}: \quad \frac{X,a,b,Y \Rightarrow c}{X,b,a,Y \Rightarrow c} \quad (\textit{Permutation})$$

$$\mathsf{C}: \quad \frac{X,a,a,Y \Rightarrow b}{X,a,Y \Rightarrow b} \quad (\textit{Contraction})$$

$$\mathsf{M}: \quad \frac{X \Rightarrow b}{X,a \Rightarrow b} \quad (\textit{Monotonicity})$$

By adding one or more of these structural rules to the Lambek calculus, we obtain stronger logics. Adding Permutation to the Lambek calculus (call the resulting system LP) results in a fragment of Linear Logic. Addition of Permutation and Contraction (LPC) gives a fragment of Relevant Logic. If all three of the above structural rules are assumed (LPCM), a fragment of Intuitionistic Logic is obtained.

Van Benthem [1, 2] studies the effect of the presence of structural rules on the recognizing power. His results include the following: LP recognizes all permutation closures of context-free languages (and probably no more); the class of LPC-recognizable languages is precisely the class of 'first-order' regular languages (see the above reference for definition); LC recognizes only regular languages. Thus, addition of structural rules generally leads to the weakening of recognizing power.

There is a different dimension in which the underlying calculus can be varied, namely the selection of connectives. The slashes in the Lambek calculus correspond to *implication* in more standard logics. Indeed, in LP, the two slashes collapse into a single connective, and in LPCM it becomes precisely the intuitionistic implication. Viewed from this perspective, it makes sense to add other kinds of connectives to the Lambek calculus, for example, analogues of *conjunction* and *disjunction*.

It turns out that there are two natural candidates for conjunction in the Lambek calculus [2]. One is *product conjunction*, which obeys the following rules:

$$(\bullet \Rightarrow)\ \frac{X, a, b, Y \Rightarrow c}{X, a \bullet b, Y \Rightarrow c} \qquad (\Rightarrow \bullet)\ \frac{X \Rightarrow a \quad Y \Rightarrow b}{X, Y \Rightarrow a \bullet b}$$

The other is *intersective conjunction*, with the following rules ($(\cap \Rightarrow)$ means two rules):

$$(\cap \Rightarrow)\ \frac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c \quad \& \quad X, b \cap a, Y \Rightarrow c} \qquad (\Rightarrow \cap)\ \frac{X \Rightarrow a \quad X \Rightarrow b}{X \Rightarrow a \cap b}$$

In the presence of Monotonicity and Contraction, these two sets of rules turn out to be merely two ways of defining the same connective. However, in the Lambek calculus, which lacks structural rules, product conjunction and intersective conjunction are two separate connectives with different properties. As a matter of fact, Lambek's original calculus in [10] included product conjunction, and intersective conjunction was already considered in [11].

The following rules define *disjunction* in the Lambek calculus:

$$(\cup \Rightarrow)\ \frac{X, a, Y \Rightarrow c \quad X, b, Y \Rightarrow c}{X, a \cup b, Y \Rightarrow c} \qquad (\Rightarrow \cup)\ \frac{X \Rightarrow a}{X \Rightarrow a \cup b \quad \& \quad X \Rightarrow b \cup a}$$

Employment of one or more of these additional connectives results in a conservative extension of the Lambek calculus, which still enjoys cut-elimination. Therefore, we do not expect a weakening of recognizing power with additional connectives; to recognize a language recognizable in the old calculus, the same type assignment works in the conservative extension.

We use an obvious notation to distinguish among various Lambek calculi[3] with different sets of connectives. Thus, $\mathsf{L}(/, \backslash, \bullet)$ is the calculus with three connectives $/$, $\backslash$, and $\bullet$ with their associated rules of inference (with no structural rules except Cut), etc.

Although logically unnatural, it is also possible to consider calculi which have only one or the other of the two kinds of introduction rules associated with certain connectives. Cut-elimination still holds in such systems. We use $\mathsf{L}(\ldots, C^-, \ldots)$ (resp. $\mathsf{L}(\ldots, C^+, \ldots)$) to denote a Lambek calculus which has only the left (resp. right) introduction rule $(C \Rightarrow)$ (resp. $(\Rightarrow C)$) for the connective C. In fact, the calculus for the classical categorial grammar of Ajdukiewicz and Bar-Hillel is an example of such a halfway system, namely $\mathsf{L}(/^-, \backslash^-)$. In the penultimate section of the paper, we will have an occasion to consider calculi $\mathsf{L}(\ldots, \cap^-, \cup^+)$.

In the sections to follow, we investigate some of the formal properties of intersective conjunction and disjunction in the Lambek calculus. To keep matters simple, we consider them one by one, deferring discussion of disjunction until intersective conjunction has been fully treated. Our main result is that intersective conjunction brings about a rather dramatic increase of the recognizing power.[4]

§2. The Lambek calculus with intersective conjunction: increased recognizing power

THEOREM I. *The Lambek calculus with intersective conjunction added* ($\mathsf{L}(/, \backslash, \cap)$) *recognizes any finite intersection of* $\mathsf{L}(/, \backslash)$*-recognizable languages.*

Since all context-free languages are $\mathsf{L}(/, \backslash)$-recognizable, it follows that $\mathsf{L}(/, \backslash, \cap)$ recognizes any finite intersection of context-free languages. The fact that the context-free languages are not closed under intersection gives us

[3]Henceforth, the term 'Lambek calculus' may be used as a generic name for calculi with no structural rules (except Cut).

[4]It is not known whether product conjunction leads to any increase in the recognizing power.

COROLLARY. $\mathsf{L}(/,\backslash,\cap)$ *recognizes some non-context-free languages.*

Proof of the Theorem. We take the case of intersection of two $\mathsf{L}(/,\backslash)$-recognizable languages. The general case is similar. Let L_1 and L_2 be languages recognized by $\mathsf{L}(/,\backslash)$-grammars G_1 and G_2, with distinguished types t_1 and t_2, respectively. The types used in the two grammars can be assumed to be distinct. We define a new $\mathsf{L}(/,\backslash,\cap)$-grammar $G_1 \cap G_2$ in the following way:

> $G_1 \cap G_2$ assigns type $a \cap b$ to symbol s if and only if G_1 assigns a to s and G_2 assigns b to s.

We discard any symbols which do not appear in both of G_1 and G_2. The distinguished type of $G_1 \cap G_2$ is $t_1 \cap t_2$.

That $L(G_1 \cap G_2) = L_1 \cap L_2$, hence the theorem, is proved if we show:

Claim. Let $a_1, \ldots, a_n$ be (conjunction-free) G_1-types, and $b_1, \ldots, b_n$ (conjunction-free) G_2-types. Then

$$a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1 \cap t_2$$

is derivable in $\mathsf{L}(/,\backslash,\cap)$ if and only if

$$a_1, \ldots, a_n \Rightarrow t_1$$

and

$$b_1, \ldots, b_n \Rightarrow t_2$$

are both derivable in $\mathsf{L}(/,\backslash)$.

Proof of the Claim.

If. It is obvious from the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{a_1, \ldots, a_n \Rightarrow t_1}{a_1 \cap b_1, a_2, \ldots, a_n \Rightarrow t_1}}{\vdots}}{a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1} \qquad \dfrac{\dfrac{\dfrac{b_1, \ldots, b_n \Rightarrow t_2}{a_1 \cap b_1, b_2, \ldots, b_n \Rightarrow t_2}}{\vdots}}{a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_2}}{a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1 \cap t_2}$$

Only if. Take any cut-free derivation of $a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1 \cap t_2$ in $\mathsf{L}(/,\backslash,\cap)$. First, we note that a sequent of the form $a \cap b \Rightarrow a \cap b$ can never appear in the derivation, except as the endsequent. If it appeared in a premise of some rule, the conclusion would necessarily involve a type which is not a subtype of any type appearing in the endsequent of the derivation, which contradicts the subformula property of cut-free derivations. In case the endsequent is of the form $a \cap b \Rightarrow a \cap b$, $a = t_1$ and $b = t_2$, and trivially, if $t_1 \cap t_2 \Rightarrow t_1 \cap t_2$ is derivable in $\mathsf{L}(/,\backslash,\cap)$, $t_1 \Rightarrow t_1$ and $t_2 \Rightarrow t_2$ are derivable in $\mathsf{L}(/,\backslash)$. Disregarding this case, we can assume axioms of the form $a \cap b \Rightarrow a \cap b$ are never used in the derivation, and all occurrences of $\cap$ in the endsequent are introduced by one of the $\cap$-introduction rules at some stage of the derivation.

It can be shown that we can 'delay' the applications of $\cap$-introduction as much as possible, producing a cut-free derivation of the same endsequent in which no $\cap$-introductions are followed by any /- or \-introductions. This is done by transforming a derivation in which a /- or \-introduction immediately follows a $\cap$-introduction to one in which they are applied in the reverse order. The following are the representative cases where a /-introduction immediately follows a $\cap$-introduction:

(1) $(\cap \Rightarrow)$ followed by $(/ \Rightarrow)$

a. $$\dfrac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \quad Z, d, V \Rightarrow e}{Z, d/c, X, a \cap b, Y, V \Rightarrow e} \rightsquigarrow \dfrac{\dfrac{X, a, Y \Rightarrow c \quad Z, d, V \Rightarrow e}{Z, d/c, X, a, Y, V \Rightarrow e}}{Z, d/c, X, a \cap b, Y, V \Rightarrow e}$$

b. $$\dfrac{X \Rightarrow a \quad \dfrac{Y, b, Z, c, V \Rightarrow e}{Y, b, Z, c \cap d, V \Rightarrow e}}{Y, b/a, X, Z, c \cap d, V \Rightarrow e} \rightsquigarrow \dfrac{\dfrac{X \Rightarrow a \quad Y, b, Z, c, V \Rightarrow e}{Y, b/a, X, Z, c, V \Rightarrow e}}{Y, b/a, X, Z, c \cap d, V \Rightarrow e}$$

c. $$\dfrac{X \Rightarrow a \quad \dfrac{Y, b, Z \Rightarrow d}{Y, b \cap c, Z \Rightarrow d}}{Y, (b \cap c)/a, X, Z \Rightarrow d}$$ does not occur because of the subformula property.

(2) $(\cap \Rightarrow)$ followed by $(\Rightarrow /)$

a. $$\dfrac{\dfrac{X, a, Y, c \Rightarrow d}{X, a \cap b, Y, c \Rightarrow d}}{X, a \cap b, Y \Rightarrow d/c} \rightsquigarrow \dfrac{\dfrac{X, a, Y, c \Rightarrow d}{X, a, Y \Rightarrow d/c}}{X, a \cap b, Y \Rightarrow d/c}$$

b. $$\dfrac{\dfrac{X, a \Rightarrow c}{X, a \cap b \Rightarrow c}}{X \Rightarrow c/(a \cap b)}$$ does not occur because of the subformula property.

(3) $(\Rightarrow \cap)$ followed by $(/ \Rightarrow)$

a. $$\dfrac{\dfrac{X \Rightarrow a \quad X \Rightarrow b}{X \Rightarrow a \cap b} \quad Y, c, Z \Rightarrow d}{Y, c/(a \cap b), X, Z \Rightarrow d}$$ does not occur because of the subformula property.

b. $$\dfrac{X \Rightarrow a \quad \dfrac{Y, b, Z \Rightarrow c \quad Y, b, Z \Rightarrow d}{Y, b, Z \Rightarrow c \cap d}}{Y, b/a, X, Z \Rightarrow c \cap d}$$

$$\rightsquigarrow \dfrac{\dfrac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, b/a, X, Z \Rightarrow c} \quad \dfrac{X \Rightarrow a \quad Y, b, Z \Rightarrow d}{Y, b/a, X, Z \Rightarrow d}}{Y, b/a, X, Z \Rightarrow c \cap d}$$

(4) $(\Rightarrow \cap)$ followed by $(\Rightarrow /)$

$$\dfrac{\dfrac{X, a \Rightarrow b \quad X, a \Rightarrow c}{X, a \Rightarrow b \cap c}}{X \Rightarrow (b \cap c)/a}$$ does not occur because of the subformula property.

The cases where a $\backslash$-introduction immediately follows a $\cap$-introduction are treated symmetrically.

It should be clear that, by performing these operations repeatedly, we can eventually obtain a cut-free derivation in which all $\cap$-introductions follow all other rules. Case (3b) increases the number of branches in the derivation, hence possibly the number of inference pairs that need to be interchanged, but the process eventually terminates. There can be at most one application of $(\Rightarrow \cap)$ throughout the process, hence the number of times the interchange in Case (3b) is performed is bounded by the height of $(\Rightarrow \cap)$ in the original derivation, for the other cases of interchange do not increase the number of inference pairs that need to be interchanged. Therefore, the proof tree will not 'grow' indefinitely. (In fact, Case (3b) never happens in the first place. This can be easily seen using the technique of numerical models described later.)

The output of this process must be a derivation with a final part that looks like the following:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{c_1,\ldots,c_n \Rightarrow t_1}}{\vdots}}{e_1,\ldots,e_n \Rightarrow t_1} \quad \dfrac{\dfrac{\dfrac{\vdots}{d_1,\ldots,d_n \Rightarrow t_2}}{\vdots}}{e_1,\ldots,e_n \Rightarrow t_2}}{\dfrac{e_1,\ldots,e_n \Rightarrow t_1 \cap t_2}{\dfrac{\vdots}{a_1 \cap b_1,\ldots,a_n \cap b_n \Rightarrow t_1 \cap t_2}}}$$

Here, each c_i and d_i are either a_i or b_i, and each e_i is a_i, b_i, or $a_i \cap b_i$. $e_1,\ldots,e_n \Rightarrow t_1$ and $e_1,\ldots,e_n \Rightarrow t_2$ are obtained from $c_1,\ldots,c_n \Rightarrow t_1$ and $d_1,\ldots,d_n \Rightarrow t_2$, respectively, by a number of (possibly zero) applications of $(\cap \Rightarrow)$. $e_1,\ldots,e_n \Rightarrow t_1 \cap t_2$ is obtained from $e_1,\ldots,e_n \Rightarrow t_1$ and $e_1,\ldots,e_n \Rightarrow t_2$ by $(\Rightarrow \cap)$. Then additional (possibly zero) applications of $(\cap \Rightarrow)$ lead to the endsequent. Note that the derivations of $c_1,\ldots,c_n \Rightarrow t_1$ and of $d_1,\ldots,d_n \Rightarrow t_2$, which occur as parts of this derivation, are derivations in $\mathsf{L}(/,\backslash)$, as well as in $\mathsf{L}(/,\backslash,\cap)$.

Now it is easy to see that it must be that $c_i = a_i$ and $d_i = b_i$, given the following (cf. [2], p. 100):

Inseparability: Let R be a relation on types such that $c\,R\,d$ iff c and d share at least one basic type. Then, $a_1,\ldots,a_m \Rightarrow b$ is derivable in one of the categorial calculi without Monotonicity only if R restricted to $\{a_1,\ldots,a_m,b\}$ is connected (i.e., $(R\cap(\{a_1,\ldots,a_m,b\} \times \{a_1,\ldots,a_m,b\}))^* = \{a_1,\ldots,a_m,b\} \times \{a_1,\ldots,a_m,b\}$).

The proof is by straightforward induction on cut-free derivations. The axioms obviously has the required property. All (normal) introduction rules preserve this property as well. (The only two of the usual structural rules that do not preserve this property are Cut and Monotonicity; thus, Inseparability holds of a wide range of so-called 'substructural' logics.)

(The fact that $c_i = a_i$ and $d_i = b_i$ implies that $e_i = a_i \cap b_i$, so that $(\Rightarrow \cap)$ is the final rule in the above derivation.)

Thus, we have been able to show that, if $a_1 \cap b_1,\ldots,a_n \cap b_n \Rightarrow t_1 \cap t_2$ is derivable in $\mathsf{L}(/,\backslash,\cap)$, then $a_1,\ldots,a_n \Rightarrow t_1$ and $b_1,\ldots,b_n \Rightarrow t_2$ are both derivable in $\mathsf{L}(/,\backslash)$, which was the 'only-if' direction of the claim.

This concludes the proof of the theorem.

The above proof assumes cut-elimination in $\mathsf{L}(/,\backslash,\cap)$, which can be proved by the standard method.

The above result does not quite answer the question about the effect of added intersective conjunction on the bidirectional Lambek calculus, since the exact limitation of the recognizing power of $\mathsf{L}(/,\backslash)$ is not known; we do not yet know that the $\mathsf{L}(/,\backslash)$-recognizable languages are not closed under intersection. However, we can see that a definite increase of the recognizing power arises when we add intersective conjunction to a unidirectional Lambek calculus ($\mathsf{L}(/)$ or $\mathsf{L}(\backslash)$), thanks to the result obtained by Buszkowski [4], which says that $\mathsf{L}(/)$ (or $\mathsf{L}(\backslash)$) recognizes precisely the context-free languages. It is clear that the above theorem retains its validity when $\mathsf{L}(/,\backslash)$ is replaced by $\mathsf{L}(/)$ (or $\mathsf{L}(\backslash)$); thus we have

THEOREM Ia. $\mathsf{L}(/,\cap)$ *and* $\mathsf{L}(\backslash,\cap)$ *recognize any finite intersection of context-free languages, hence some non-context-free languages.*

Example. By Theorems I and Ia, $L = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 1\} = \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^m \mid n \geq 1, m \geq 1\} \cap \{\mathtt{a}^m\mathtt{b}^n\mathtt{c}^n \mid m \geq 1, n \geq 1\}$ is recognized by $\mathsf{L}(/,\backslash,\cap)$, $\mathsf{L}(/,\cap)$, and $\mathsf{L}(\backslash,\cap)$, but not by $\mathsf{L}(/)$ or $\mathsf{L}(\backslash)$.

Theorem I can be strengthened to the following form:

THEOREM II. *The* $\mathsf{L}(/,\backslash,\cap)$*-recognizable languages are closed under intersection.*

Proof. The same method used in Theorem I can be used here. The theorem follows from the strengthened version of the Claim in the proof of Theorem I:

Claim. Let $a_1,\ldots,a_n,t_1$ and $b_1,\ldots,b_n,t_2$ be two sequences of $\mathsf{L}(/,\backslash,\cap)$-types based on disjoint sets A and B of basic types, respectively. Then

$$a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1 \cap t_2$$

is derivable in $\mathsf{L}(/,\backslash,\cap)$ if and only if

$$a_1, \ldots, a_n \Rightarrow t_1$$

and

$$b_1, \ldots, b_n \Rightarrow t_2$$

are both derivable in $\mathsf{L}(/,\backslash,\cap)$.

To prove the above claim, what we do is to construct, given a cut-free derivation of $a_1 \cap b_1, \ldots, a_n \cap b_n \Rightarrow t_1 \cap t_2$, one in which all applications of $\cap$-introduction *which correspond to the main connectives of the formulas in the endsequent* follow all other rule applications. (The bulk of the present proof consists in replacing 'applications of $\cap$-introduction rules' in the proof of Theorem I by 'applications of $\cap$-introduction rules which introduce the main connectives of the formulas of the endsequent'.) Interchange of $\cap$-introduction and introduction of one of the slashes is possible in the precise same way as before. In particular, those cases ruled out by the subformula property still remain to be so ruled out, since now we are trying to 'delay' only those applications of $\cap$-rules which introduce the main connectives of the formulas in the endsequent, which cannot lie within the scope of any connectives in the course of a cut-free derivation. However, we now have some new cases to deal with as well:

(5) $(\cap \Rightarrow)$ followed by $(\cap \Rightarrow)$

$$\dfrac{\dfrac{X, a, Y, c, Z \Rightarrow e}{X, a \cap b, Y, c, Z \Rightarrow e}}{X, a \cap b, Y, c \cap d, Z \Rightarrow e} \quad \rightsquigarrow \quad \dfrac{\dfrac{X, a, Y, c, Z \Rightarrow e}{X, a, Y, c \cap d, Z \Rightarrow e}}{X, a \cap b, Y, c \cap d, Z \Rightarrow e}$$

(6) $(\Rightarrow \cap)$ followed by $(\cap \Rightarrow)$

$$\dfrac{\dfrac{X, a, Y \Rightarrow c \qquad X, a, Y \Rightarrow d}{X, a, Y \Rightarrow c \cap d}}{X, a \cap b, Y \Rightarrow c \cap d} \quad \rightsquigarrow \quad \dfrac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \qquad \dfrac{X, a, Y \Rightarrow d}{X, a \cap b, Y \Rightarrow d}}{X, a \cap b, Y \Rightarrow c \cap d}$$

(7) $(\Rightarrow \cap)$ followed by $(\Rightarrow \cap)$

$$\dfrac{\dfrac{X \Rightarrow a \quad X \Rightarrow b}{X \Rightarrow a \cap b} \quad X \Rightarrow c}{X \Rightarrow (a \cap b) \cap c}$$ need not be treated because of the subformula property.

In addition, there is one *non-permutable* case:

(8) $(\cap \Rightarrow)$ followed by $(\Rightarrow \cap)$

$$\dfrac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \quad X, a \cap b, Y \Rightarrow d}{X, a \cap b, Y \Rightarrow c \cap d}$$

Thus, the present theorem requires a more delicate argument than Theorem I. We proceed in the following fashion. First, we try to delay the introduction of the main connectives of the formulas in the endsequent as much as possible, by performing the operations of interchange in (1)–(7) (plus those other cases which are essentially the same) to the given derivation repeatedly, until no more such operation is possible. We look at the resulting derivation. Call a situation like (8) where $a \cap b = a_i \cap b_i$ and $c \cap d \neq t_1 \cap t_2$ an *unpermuted situation*. If no unpermuted situation obtains in the resulting derivation, we are done. If there are such situations, we take a highest one. We suppose this looks exactly like (8), for illustration. No formula in the endsequent originates in an axiom, except when the endsequent itself is an axiom, because of the subformula property and the given condition on the distribution of basic types in the endsequent. Hence the occurrence of $a \cap b$ in the right premise (as well as that in the left) must be introduced by an application of $(\cap \Rightarrow)$. There can be no rule applications between that application of $(\cap \Rightarrow)$ and the application of $(\Rightarrow \cap)$ in (8), except possibly some applications of $(\cap \Rightarrow)$ which introduce the main connectives of the endsequent, since otherwise the required kind of interchange would be possible or there would be an unpermuted situation above the one chosen ($t_1 \cap t_2$ cannot have been introduced above). In any case, an adjacent pair of applications of $(\cap \Rightarrow)$ can always be interchanged according to (5), so we can bring it about that the right premise in (8) is itself the conclusion of the application of $(\cap \Rightarrow)$ which introduces $a \cap b$. Therefore, we have a situation that looks like one of the following:

(a) $$\dfrac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \quad \dfrac{X, a, Y \Rightarrow d}{X, a \cap b, Y \Rightarrow d}}{X, a \cap b, Y \Rightarrow c \cap d}$$

(b) $$\dfrac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \quad \dfrac{X, b, Y \Rightarrow d}{X, a \cap b, Y \Rightarrow d}}{X, a \cap b, Y \Rightarrow c \cap d}$$

The first case is permutable:

$$\dfrac{\dfrac{X, a, Y \Rightarrow c \quad X, a, Y \Rightarrow d}{X, a, Y \Rightarrow c \cap d}}{X, a \cap b, Y \Rightarrow c \cap d}$$

After this permutation, the whole process is repeated. As for the second case, we can conclude that it cannot obtain, since we are led to a contradiction from the fact that $c \cap d$ is either a pure A-type or a pure B-type. Suppose $c \cap d$ is a pure A-type, the other case being symmetric. Then d is a pure A-type and the sequent $X, b, Y \Rightarrow d$ can be shown to be underivable. If all types in X and Y are either pure A-types or pure B-types, then the sequent is shown to be underivable by Inseparability Lemma. If X, Y contains some formulas from the endsequent, then the main connectives of all those formulas must be introduced immediately above, because we are looking at a highest non-permutable situation involving an application of ($\cap \Rightarrow$) that introduces one of the main connectives in the endsequent. (Note that $t_1 \cap t_2$ cannot have been introduced above.) Therefore, there must be a sequent $X', b, Y' \Rightarrow d$ above $X, b, Y \Rightarrow d$ such that all formulas in X' and Y' are either pure A-types or pure B-types. Then this sequent must be underivable by Inseparability.

The rest of the proof runs exactly the same as the proof of Theorem I.

Clearly, from the proof of Theorem II, we also have

THEOREM IIa. *The* $\mathsf{L}(/, \cap)$*-recognizable languages and the* $\mathsf{L}(\backslash, \cap)$*-recognizable languages are closed under intersection.*

It can also be generalized to the case where product conjunction is present as well.[5]

THEOREM IIb. *The* $\mathsf{L}(/, \backslash, \bullet, \cap)$*-recognizable languages are closed under intersection.*

Proof. Exactly as before, noting the following new cases of interchange:

(9) ($\cap \Rightarrow$) followed by ($\bullet \Rightarrow$)

$$\frac{\dfrac{X, a, Y, c, d, Z \Rightarrow e}{X, a \cap b, Y, c, d, Z \Rightarrow e}}{X, a \cap b, Y, c \bullet d, Z \Rightarrow e} \quad \rightsquigarrow \quad \frac{\dfrac{X, a, Y, c, d, Z \Rightarrow e}{X, a, Y, c \bullet d, Z \Rightarrow e}}{X, a \cap b, Y, c \bullet d, Z \Rightarrow e}$$

(10) ($\cap \Rightarrow$) followed by ($\Rightarrow \bullet$)

$$\frac{\dfrac{X, a, Y \Rightarrow c}{X, a \cap b, Y \Rightarrow c} \quad Z \Rightarrow d}{X, a \cap b, Y, Z \Rightarrow c \bullet d} \quad \rightsquigarrow \quad \frac{\dfrac{X, a, Y \Rightarrow c \quad Z \Rightarrow d}{X, a, Y, Z \Rightarrow c \bullet d}}{X, a \cap b, Y, Z \Rightarrow c \bullet d}$$

(11) ($\Rightarrow \cap$) followed by ($\bullet \Rightarrow$)

$$\frac{\dfrac{X, a, b, Y \Rightarrow c \quad X, a, b, Y \Rightarrow d}{X, a, b, Y \Rightarrow c \cap d}}{X, a \bullet b, Y \Rightarrow c \cap d} \quad \rightsquigarrow \quad \frac{\dfrac{X, a, b, Y \Rightarrow c}{X, a \bullet b, Y \Rightarrow c} \quad \dfrac{X, a, b, Y \Rightarrow d}{X, a \bullet b, Y \Rightarrow d}}{X, a \bullet b, Y \Rightarrow c \cap d}$$

(The cases excluded by the subformula property are omitted.)

Remark. The method of proof of Theorems I, II, and IIb is reminiscent of Curry's [7] and Kleene's [9] results about the permutability of inferences in the classical and intuitionistic calculi. It demonstrates some partial permutability theorems for the Lambek calculus. At the same time, it reveals additional obstacles to proving such results. For example, the case of ($\cap \Rightarrow$) followed by ($\Rightarrow \cap$) is a case of non-permutability new to the Lambek calculus. Kleene's permutability theorem depends on freewheeling use of Thinning and Contraction, which is not available in the case of the Lambek calculus, while Curry's theorem depends on a particular formulation of rules (parametric formulas occur in all premises and the conclusion), which is too restrictive to be useful in the case of the Lambek calculus. (Curry's theorem is also more limited in the scope of application.) Here is another respect in which the lack of structural rules brings in additional subtlety to logic.

[5] Disjunction may also be present, though it adds two new non-permutable cases.

In the following discussion, let $\overline{C}$ stand uniformly for one or both of the slashes (/, \; or /; or \). Theorems I and Ia give a certain lower bound on the class of $\mathsf{L}(\overline{C}, \cap)$-recognizable languages—namely, the class of all finite intersections of $\mathsf{L}(\overline{C})$-recognizable languages. In light of Theorems II and IIa, one might ask whether that lower bound in fact exactly characterizes the class of $\mathsf{L}(\overline{C}, \cap)$-recognizable languages, i.e., whether the effect of adding $\cap$ to $\mathsf{L}(\overline{C})$ on the class of recognizable languages amounts to closing the original class under intersection. The answer is negative for $\overline{C} = /$ or \, and it would be natural to conjecture the same for $\overline{C} = /, \backslash$.

THEOREM III. *The class of* $\mathsf{L}(\overline{C}, \cap)$*-recognizable languages properly includes the class of finite intersections of* $\mathsf{L}(\overline{C})$*-recognizable languages, for* $\overline{C} = /$ *or* \.

Proof. We call a relation $R \in \Sigma \times \Sigma'$, where Σ and Σ' are two finite alphabets, a *symbol-to-symbol substitution from* Σ *to* Σ'. A function h from Σ to Σ' is a *symbol-to-symbol homomorphism*, and is regarded as a special kind of symbol-to-symbol substitution. Symbol-to-symbol substitutions are naturally extended to the level of strings (and to the level of languages), by $\epsilon\, R\, \epsilon$, $xa\, R\, y$ iff $y = x'a'$ for some x' and a' such that $x\, R\, x'$, $a\, R\, a'$. We let $LR = \{y \mid \exists x \in L\ x\, R\, y\}$. A family of languages $\mathcal{L}$ is said to be closed under symbol-to-symbol substitution if $L \in \mathcal{L}$ implies $LR \in \mathcal{L}$ for any symbol-to-symbol substitution R. The following two lemmas establish the theorem.

LEMMA 1. *For any calculus* $\mathsf{L}(C_1, \ldots, C_m)$*, the class of* $\mathsf{L}(C_1, \ldots, C_m)$*-recognizable languages is closed under symbol-to-symbol substitution.*

Proof. For a categorial grammar G, let T be the set of types assigned to some symbol by G. T is finite. Then the assignment of G can be regarded as a symbol-to-symbol substitution R from Σ (the alphabet of G) to T. Let $L(T, t) = \{X \in T^* \mid X \Rightarrow t \text{ is derivable}\}$, where t is the distinguished type of G and the derivability is the derivability in the calculus underlying G. Then $L(G) = L(T, t)R^{-1}$. Let S be a symbol-to-symbol substitution from Σ to some alphabet Σ'. Then $L(G)S = (L(T, t)R^{-1})S = L(T, t)R^{-1} \circ S$. $R^{-1} \circ S$ is a symbol-to-symbol substitution from T to Σ', and $(R^{-1} \circ S)^{-1} = S^{-1} \circ R$ is the assignment for a grammar G' recognizing $L(G)S$.

LEMMA 2. *The class of finite intersections of context-free languages is not closed under symbol-to-symbol homomorphism.*

Proof [12]. Consider the following context-free languages:

$$L_0 = \{a^i b^i \mid i \geq 1\},$$
$$L_1 = L_0^*,$$
$$L_2 = \{b^i a^i \mid i \geq 1\},$$
$$L_3 = \bigcup_{n \geq 1} a^n L_2^{n-1} b^+.$$

$L_1 \cap L_3 = \{(a^n b^n)^n \mid n \geq 1\}$. A symbol-to-symbol homomorphism h such that $h(a) = h(b) = a$ maps $L_1 \cap L_3$ to $\{a^{2n^2} \mid n \geq 1\}$, which is not a regular language and thus cannot be expressed as a finite intersection of context-free languages, since all one-symbol context-free languages are regular and the regular languages are closed under intersection.

Just as in the case of $\mathsf{L}(/,\backslash)$, the limitation of the recognizing power of $\mathsf{L}(\overline{C},\cap)$ has not been obtained, except for the decidability result which follows from cut-elimination. It should be clear, however, that $\mathsf{L}(\overline{C},\cap)$ cannot recognize all recursive languages, since the $\mathsf{L}(\overline{C},\cap)$-grammars can be effectively enumerated.

Question. Do the $\mathsf{L}(\overline{C},\cap)$-recognizable languages form a (proper) subclass of the class of context-sensitive languages?

§3. Single type assignment by means of intersective conjunction

The method of proof of Theorem I suggests that assignment of a single conjunctive type $a \cap b$ to symbol s can in a way mimic assignment of two types a and b to s. The following theorem says that we can indeed get by with only single type assignment using intersective conjunction for a wide range of $\mathsf{L}(/,\backslash,\bullet,\cap)$-recognizable languages.[6]

THEOREM IV. *Any finite intersection of* $\mathsf{L}(/,\backslash,\bullet)$*-recognizable languages can be recognized by* $\mathsf{L}(/,\backslash,\bullet,\cap)$ *via single type assignment. (Similarly with* $/,\backslash,\bullet$ *replaced by* $/,\backslash$ *or* $/$ *or* $\backslash$*.)*

Proof. Given any $\mathsf{L}(/,\backslash,\bullet)$-grammar G, construct a $\mathsf{L}(/,\backslash,\bullet,\cap)$-grammar G' as follows. If $a_1,\ldots,a_m$ are the types assigned to symbol s by G, then $a_1 \cap \cdots \cap a_m$ is the (only) type assigned to s by G'. The distinguished type remains the same. The fact that G and G' recognize the same language can be seen using the by-now-familiar style of proof analysis. Then we only need to observe that the $\mathsf{L}(/,\backslash,\bullet,\cap)$-grammar $G_1 \cap G_2$ constructed according to the recipe found in the proof of Theorem I is a single type assignment grammar if G_1 and G_2 are.

Whether single type assignment suffices for the general case of $\mathsf{L}(/,\backslash,\bullet,\cap)$-recognizable languages is less clear. The above construction does not work for a case like the following:

G	Symbol	0	1
	Types assigned	p, q	$(p \cap q)\backslash t$
G'	Symbol	0	1
	Types assigned	$p \cap q$	$(p \cap q)\backslash t$

G recognizes the empty set, whereas G' recognizes the singleton set consisting of 01.

To turn the table around, I have been unable to prove that multiple type assignment is essential for some $\mathsf{L}(/,\backslash,\bullet)$-recognizable languages, although it is easy to see that it is, if we confine ourselves to grammars whose distinguished type is *basic*.

Example. The regular language 00* cannot be recognized by a $\mathsf{L}(/,\backslash,\bullet)$-grammar with single type assignment and a basic distinguished type. This can be seen using the technique of *primitive type counts* [2]. For each basic type x, define the x-count $\#_x(a)$ of any type a as follows:

$$\#_x(x) = 1,$$
$$\#_x(y) = 0, \qquad \text{for all basic types } y \text{ distinct from } x,$$
$$\#_x(b/a) = \#_x(a\backslash b) = \#_x(b) - \#_x(a),$$
$$\#_x(a \bullet b) = \#_x(a) + \#_x(b).$$

[6] The idea of using $\cap$ to achieve single type assignment is already present in [11].

The x-count of a sequence X of types is just the sum of those of the types in X. It can be seen that for each derivable sequent $X \Rightarrow a$ in $\mathsf{L}(/, \backslash, \bullet)$, all primitive type counts must be equal on both sides of the sequent [2]. By this invariance, any $\mathsf{L}(/, \backslash, \bullet)$-grammar G with a basic distinguished type t which recognizes $\mathtt{0}\mathtt{0}^*$ must assign a type a to $\mathtt{0}$ which has $\#_t(a) = 1$, to recognize $\mathtt{0}$. Then such a grammar must assign some other type to $\mathtt{0}$ as well, to recognize $\mathtt{0}^n$ for $n \geq 2$. However, if a complex type t/t is allowed to be the distinguished type, a single type assignment grammar can recognize $\mathtt{0}\mathtt{0}^*$, by assigning t/t to $\mathtt{0}$.

Remark 1. Some authors take the distinguished type to be basic in the definition of categorial grammars (cf. [4]). This is not a severe restriction, however, in terms of the complexity of the language recognized. For, if L is recognized by $\mathsf{L}(/, \ldots)$ via a grammar G with some complex distinguished type x, $\mathtt{z}L = \{\mathtt{z}w \mid w \in L\}$ ($\mathtt{z}$ is a symbol not occurring in L) is recognized by the same calculus via a grammar G' with a (new) basic distinguished type t, by assigning t/x to $\mathtt{z}$. L is obtained from $\mathtt{z}L$ by *limited erasing*, so any 'natural' family of languages (*trio*, cf. [8]) which contains $\mathtt{z}L$ also contains L. For example, if L is not context-free, then neither is $\mathtt{z}L$. Therefore, our result about the effect of added $\cap$ on the recognizing power can be translated in terms of categorial grammars with basic distinguished types. In particular, some non-context-free languages are $\mathsf{L}(/, \ldots, \cap)$-recognizable via basic distinguished types. Using complex distinguished types allows more smooth formulation of the kind of results obtained in the present paper.

Remark 2. As [5] notes, any language recognized by some calculus via multiple type assignment is the image under a symbol-to-symbol homomorphism of a language recognized by the same calculus via single type assignment. Thus, the languages recognized via single type assignment can in a sense encode arbitrary languages recognized via multiple type assignment, and, essentially, the complexity of the latter is already present in the former. A remark concerning natural families of languages analogous to that made in the preceding paragraph applies here as well.

We now proceed to give certain general properties of the calculus $\mathsf{L}(/, \backslash, \bullet, \cap)$ itself.

§4. Numerical models for the Lambek calculus

We give a certain rough semantics for the Lambek calculus in terms of *numerical models* [2].[7] The semantic tool here is sets of vectors in N^k, for a convenient choice of k. A numerical model N assigns to each basic type a a subset N_a of N^k as its 'denotation'. The denotation for complex types is computed in the following way:

$$N_{b/a} = N_{a\backslash b} = N_a \to N_b = \{x \mid \forall y \in N_a \; x + y \in N_b\},$$
$$N_{a\bullet b} = N_a + N_b = \{x + y \mid x \in N_a, y \in N_b\},$$
$$N_{a\cap b} = N_a \cap N_b.$$

A sequent $a_1, \ldots, a_n \Rightarrow b$ is numerically valid if for all numerical models N, $N_{a_1} + \cdots + N_{a_n} \subseteq N_b$. By straightforward induction on derivations, we have the following:

PROPOSITION. $\mathsf{L}(/, \backslash, \bullet, \cap)$ *is sound with respect to numerical models.*

Validity in numerical models is only a rough approximation of derivability in $\mathsf{L}(/, \backslash, \bullet, \cap)$, since the former does not distinguish a sequent from another obtainable from it by permuting

[7]Semantics in terms of language models, which was mentioned early on in the paper, also admits of natural extension to $\mathsf{L}(/, \backslash, \bullet, \cap)$. However, the completeness of even $\mathsf{L}(/, \backslash, \bullet)$ with respct to language models has not been obtained.

the antecedent. Nevertheless, numerical models turn out to be very useful in certain cases.[8] For example, here is a generalization of the earlier Inseparability Lemma:

THEOREM V. *Let A and B be disjoint sets of basic types. Then any sequent $X, c, Y \Rightarrow d$ which meets the following conditions is underivable in* $\mathsf{L}(/, \backslash, \bullet, \cap)$*:*

1. *All subtypes of the form b/a or $a\backslash b$ which occur in the sequent are either pure A-types or pure B-types.*
2. *c is either a pure A-type or a type constructed from a pure A-type and other types using $\bullet$ only.*
3. *d is either a pure B-type or a type constructed from a pure B-type and other types using $\cap$ only.*

Proof. Let N be a numerical model which assigns N to all basic types in A and $\{0\}$ to all basic types in B. Then it is easy to check the following:

- N assigns N to all pure A-types.
- N assigns $\{0\}$ to all pure B-types.
- N assigns either N or $\{0\}$ to all types in $X, c, Y \Rightarrow d$.
- N assigns N to c.
- N assigns $\{0\}$ to d.

N clearly invalidates $X, c, Y \Rightarrow d$.

It is now clear that Case (3b) in the proof of Theorem I never occurs, since the final sequent of (3b) meets the conditions of Theorem V. This theorem also provides an alternative treatment of case (b) in the proof of Theorem II.

Remark. The proof of Theorem V suggests a general perspective. Any finite class of subsets of N^k which is closed under the operations of $\rightarrow$, $+$, $\cap$ generates certain 'many-valued truth tables'. For example, what the proof of Theorem V used is the following '3-valued logic':

$\rightarrow$	N	$\{0\}$	$\emptyset$
N	N	$\emptyset$	$\emptyset$
$\{0\}$	N	$\{0\}$	$\emptyset$
$\emptyset$	N	N	N

$+$	N	$\{0\}$	$\emptyset$
N	N	N	$\emptyset$
$\{0\}$	N	$\{0\}$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

$\cap$	N	$\{0\}$	$\emptyset$
N	N	$\{0\}$	$\emptyset$
$\{0\}$	$\{0\}$	$\{0\}$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

[8] In fact, in the scope of pure implicational formulas, [2] conjectures that the undirected Lambek calculus (Lambek calculus with the structural rule of Permutation) is complete with respect to numerical models. To be precise, the calculus complete with respect to numerical models must be as strong as $\mathsf{L^*P}$—the Lambek calculus which allows empty antecedents, with Permutation. The latter is sound with respect to numerical models, where empty antecedents are interpreted as $\{0\}$, and stronger than LP—the Lambek calculus which disallows empty antecedents, with Permutation. For instance, numerically valid $(a \rightarrow a) \rightarrow b \Rightarrow b$ is derivable in $\mathsf{L^*P}$, but not in LP. (With Permutation, / and \ collapse to a single connective, which we represent by $\rightarrow$ here.) The conjecture might naturally be extended to $\mathsf{L^*P}$ enriched with additional connectives considered in this paper.

Combination of N and $\{0\}$ always yields N or $\{0\}$, except for $\mathsf{N} \rightarrow \{0\}$, the fact utilized in the proof of Theorem V. The 'subtables' of the above restricted to N and $\emptyset$ are identical to classical 2-valued valuation, with two conjunctions collapsing to one. [2] already makes use of this fact in showing certain classically invalid sequents to be underivable in LP. The following is an instance of an intuitionistically valid sequent which can be shown to be underivable in the Lambek calculus by the above table for $\rightarrow$:

$$\begin{array}{ccccccccc} t & \Rightarrow & (& e & \backslash & e &) & \backslash & t \\ \{0\} & & & \mathsf{N} & & \mathsf{N} & & & \{0\} \\ & & & & \mathsf{N} & & & & \\ & & & & & & & \emptyset & \end{array}$$

This sequent escapes the check of the primitive type counts mentioned earlier.

Finite numerical models. As Johan van Benthem has pointed out to me, it is possible to replace the structure $\langle \mathsf{N}, + \rangle$ in the numerical modelling by a finite structure $\langle \mathsf{N}_m, +_m \rangle$, for $m \neq 0$, where $\mathsf{N}_m = \{[n]_m \mid n \in \mathsf{N}\}$ is the collection of *congruence classes modulo* m and $+_m$ is the associated operation of addition on those congruence classes. In particular, in the above 3-valued tables, N and $\{0\}$ can be replaced by $\{[0]_2, [1]_2\}$ and $\{[0]_2\}$, respectively.

Question. Does the validity with respect to numerical models based on N (or Z; see below) coincide with the validity with respect to all finite numerical models based on congruence classes modulo m for $m \neq 0$? With m restricted to prime numbers?

§5. Use of count invariance for the Lambek calculus with intersective conjunction

Useful as they are, numerical models do not have the convenience of primitive type counts. Which numerical model to use to refute a sequent at hand seems to be a matter of ingenuity. Is there a way to generalize count invariance to $\mathsf{L}(/, \backslash, \bullet, \cap)$?

Numerical modelling itself, when extended to include negative integers in the domain in addition to elements of N, provides one candidate [2]. Here we focus on singleton sets of vectors in Z^k, where k is the number of basic types we are dealing with and the i-th coordinate represents a primitive type count with respect to the i-th basic type. Then assigning the i-th basic type the singleton set of the vector each of whose coordinate is zero except the i-th, which receives the value 1, this special numerical model simulates primitive type counts in the scope of $\mathsf{L}(/, \backslash, \bullet)$-formulas, as can be easily checked. As it stands, however, this method is not quite successful in the presence of $\cap$, since in case a and b have different type counts, $a \cap b$ will receive the value $\emptyset$ in this model. Then this model cannot reject a sequent like $p \cap q \Rightarrow t$, which, intuitively, should be rejected on the basis of some kind of type count system.

There might be a way to get around this and 'embed' an adequate type count system for $\mathsf{L}(/, \backslash, \bullet, \cap)$ in numerical models, but let us seek a different direction. What we will do is in a sense a combination of proof search and count check. Roughly, given a $\mathsf{L}(/, \backslash, \bullet, \cap)$-sequent $X \Rightarrow a$ we eliminate all occurrences of $\cap$ in it and get a $\mathsf{L}(/, \backslash, \bullet)$-sequent $X' \Rightarrow a'$ such that if $X \Rightarrow a$ is derivable, then $X' \Rightarrow a'$ is also derivable. Then we can test count invariance on the latter sequent. If the count check rejects it, then the original sequent must be rejected. Our permutability result already enables us to eliminate those occurrences of $\cap$ which do not lie within the scope of $/$, $\backslash$, or $\bullet$, but here we want to eliminate all occurrences of $\cap$. Theorems VI and VII below provide a way to do it.

We understand *positive* and *negative occurrences* of subformulas in a formula in the usual sense. Thus, in b, b/a, $a\backslash b$, $b \bullet c$, and $b \cap c$, the indicated occurrences of b and c are positive, and the indicated occurrences of a are negative. If b occurs positively in a, and c occurs positively (resp. negatively) in b, then c occurs positively (resp. negatively) in a. If b occurs negatively in a, and c occurs positively (resp. negatively) in b, then c occurs negatively (resp. positively) in a. Moreover, we say an antecedent formula of a sequent occurs negatively in the sequent, and the succedent formula positively. Subformulas which occur positively (resp. negatively) in an antecedent formula of a sequent occur negatively (resp. positively) in the sequent. Subformulas which occur positively (resp. negatively) in the succedent formula of a sequent occur positively (resp. negatively) in the sequent. We indicate positive and negative occurrences of a subformula b in a formula or a sequent by b^+ and b^-, respectively. If a formula a (or a sequent $\mathcal{S}$) has a specific positive occurrence of b in it, we may use $a(b^+)$ (or $\mathcal{S}(b^+)$) to denote a (or $\mathcal{S}$), and use $a(c^+)$ (or $\mathcal{S}(c^+)$) to denote the result of replacing the indicated occurrence of b in $a(b^+)$ (or $\mathcal{S}(b^+)$) by c. Likewise for negative occurrences. In case $b = c\,C\,d$, we may write $c\,C^+\,d$ and $c\,C^-\,d$ for b^+ and b^-, respectively.

We are now ready to state our theorems.

THEOREM VI. *If $\mathcal{S}(a \cap^+ b)$ is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap)$, then so are $\mathcal{S}(a^+)$ and $\mathcal{S}(b^+)$.*

Proof. Positive subformula occurrences in a sequent are positive subformula occurrences in its succedent formula and negative subformula occurrences in its antecedent formulas. Since $a \cap b \Rightarrow a$ and $a \cap b \Rightarrow b$ are derivable in $\mathsf{L}(/, \backslash, \bullet, \cap)$, the theorem follows from the following, using Cut:

Fact. The derivability of $c \Rightarrow d$ implies the derivability of $e(c^+) \Rightarrow e(d^+)$ and $e(d^-) \Rightarrow e(c^-)$.

The proof is by straightforward induction on the complexity of e.

THEOREM VII. *Suppose $\mathcal{S}(a \cap^- b)$ has no positive occurrence of a subformula of the form $c \cap d$. Then if $\mathcal{S}(a \cap^- b)$ is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap)$, then either $\mathcal{S}(a^-)$ or $\mathcal{S}(b^-)$ is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap)$.*

Proof. Take a cut-free derivation $\mathcal{F}$ of $\mathcal{S}(a \cap^- b)$. Tracing back the 'ancestors' of subformula occurrences in $\mathcal{S}(a \cap^- b)$, it can be seen that a positive (resp. negative) occurrence of a subformula $c\,C\,d$ in $\mathcal{S}(a \cap^- b)$ originates either as (i) a positive (resp. negative) occurrence in an axiom; (ii) the principal formula of an application of $(\Rightarrow C)$ (resp. $(C \Rightarrow)$); or (iii) a positive (resp. negative) occurrence in the conclusion of an application of $(\cap \Rightarrow)$, as a subformula of the 'irrelevant conjunct' of its principal formula $e \cap f$. There must be a unique origin, since there is no application of $(\Rightarrow \cap)$, which is the only rule which has an effect of 'implicit contraction' of parametric formulas. In the case of the indicated occurrence of $a \cap^- b$ in $\mathcal{S}(a \cap^- b)$, case (i) is excluded, since there are no positive occurrences of $a \cap b$ in $\mathcal{S}(a \cap^- b)$.

We modify $\mathcal{F}$ by replacing all ancestors of the indicated occurrence of $a \cap^- b$ (including itself) by a and by b, obtaining two derivation-like figures, $\mathcal{F}_a$ and $\mathcal{F}_b$. Each rule application in $\mathcal{F}$ in which $a \cap^- b$ occurs as a subformula of a parametric formula or a side formula (including the 'irrelevant conjunct') remains an application of the same rule after this transformation. Therefore, if case (iii) obtains for $a \cap^- b$, $\mathcal{F}_a$ and $\mathcal{F}_b$ are legitimate derivations of $\mathcal{S}(a^-)$ and $\mathcal{S}(b^-)$. In case (ii), the only fault of $\mathcal{F}_a$ and $\mathcal{F}_b$ as derivations is that what used to be an application of $(\cap \Rightarrow)$ is no more an instance of inference:

$$\mathcal{F}\colon \dfrac{\dfrac{\vdots}{X, x, Y \Rightarrow z}}{\dfrac{X, a \cap b, Y \Rightarrow z}{\vdots}}(\cap \Rightarrow) \quad \rightsquigarrow \quad \mathcal{F}_a\colon \dfrac{\dfrac{\vdots}{X, x, Y \Rightarrow z}}{\dfrac{X, a, Y \Rightarrow z}{\vdots}} \qquad \mathcal{F}_b\colon \dfrac{\dfrac{\vdots}{X, x, Y \Rightarrow z}}{\dfrac{X, b, Y \Rightarrow z}{\vdots}}$$

$x = a$ or b. If $x = a$, we take $\mathcal{F}_a$. We simply delete one of the occurrences of $X, a, Y \Rightarrow z$ in the portion of $\mathcal{F}_a$ depicted above. It should be clear that the result is a cut-free derivation of $\mathcal{S}(a^-)$. If $x = b$, we proceed symmetrically.

These theorems suggest the following generalized count check procedure. Given a $\mathsf{L}(/, \backslash, \bullet, \cap)$-sequent $\mathcal{S}$, we first obtain a set of sequents $\mathcal{S}_1, \ldots, \mathcal{S}_m$ by eliminating all positive occurrences of subformulas of the form $a \cap b$ in $\mathcal{F}$, replacing them by a or by b. (If there are k positive occurrences of subformulas of the form $a \cap b$, then m can be as large as 2^k.) Then for each $\mathcal{S}_i$, we replace all negative occurrences of subformulas of the form $c \cap d$ by c or by d, obtaining a set of $\cap$-free sequents $\mathcal{S}_{i,1}, \ldots, \mathcal{S}_{i,n_i}$. (If there are l negative occurrences of subformulas of the form $a \cap b$ in the original sequent, each n_i can be as large as 2^l.) Then we check each $\mathcal{S}_{i,j}$ against count invariance. The sequent $\mathcal{S}$ is accepted if for all i, one of $\mathcal{S}_{i,1}, \ldots, \mathcal{S}_{i,n_i}$ is accepted by the original count check, and rejected otherwise.

In the above procedure, a large number of similar-looking sequents must be constructed and at the final step each of them must be checked separately against count invariance. Here is a different, 'bottom-up' method of calculation:

Generalized primitive type count. Given a sequent $X \Rightarrow d$, we assign each subformula occurrence a in it a finite set $\#(a)$ of finite sets of vectors in Z^k, where k is the number of basic types involved in the sequent. We first mark each occurrence of a subformula of the form $a \cap b$ with its polarity, indicating whether it occurs positively or negatively in the given sequent. $\#(a)$ is calculated as follows:

$$\begin{aligned}
&\#(e_i) = \{\{\langle 0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0\rangle\}\}, && \text{for the } i\text{-th basic type } e_i,\\
&\#(b/a) = \#(a\backslash b) = \#(b) - \#(a)\\
&\qquad = \{B - A \mid B \in \#(b), A \in \#(a)\}\\
&\qquad = \{\{y - x \mid y \in B, x \in A\} \mid B \in \#(b), A \in \#(a)\},\\
&\#(a \bullet b) = \#(a) + \#(b)\\
&\qquad = \{A + B \mid A \in \#(a), B \in \#(b)\}\\
&\qquad = \{\{x + y \mid x \in A, y \in B\} \mid A \in \#(a), B \in \#(b)\},\\
&\#(a \cap^+ b) = \#(a) \cup \#(b),\\
&\#(a \cap^- b) = \{A \cup B \mid A \in \#(a), B \in \#(b)\}.
\end{aligned}$$

The antecedent $X = c_1, \ldots, c_n$ receives the value $\#(X) = \#(c_1) + \cdots + \#(c_n)$. $\#(a)$ is interpreted like a 'conjunction of disjunctions'. The sequent $X \Rightarrow d$ is accepted if for all A in $\#(X)$ and for all B in $\#(d)$, $A \cap B \neq \emptyset$, and rejected otherwise. The equivalence of the earlier method and the present one can be checked, and the 'soundness' of the latter may be proved directly by induction on cut-free derivations.

Example. The following sequent is shown to be underivable, by our generalized count invariance:

$$(t/[(e\backslash t) \cap^+ (e\backslash(t/e))]) \cap^- (t/t),\ (e\backslash(t/e)) \cap^- ((t/t)\backslash e) \ \Rightarrow\ t \cap^+ e$$

Calculation is as follows (the first coordinate represents e-count and the second t-count):

$$\begin{aligned}
&\#((e\backslash t)\cap^{+}(e\backslash(t/e))) = \{\{\langle -1,1\rangle\},\{\langle -2,1\rangle\}\},\\
&\#(t/[(e\backslash t)\cap^{+}(e\backslash(t/e))]) = \{\{\langle 1,0\rangle\},\{\langle 2,0\rangle\}\},\\
&\#((t/[(e\backslash t)\cap^{+}(e\backslash(t/e))])\cap^{-}(t/t)) = \{\{\langle 1,0\rangle,\langle 0,0\rangle\},\{\langle 2,0\rangle,\langle 0,0\rangle\}\},\\
&\#((e\backslash(t/e))\cap^{-}((t/t)\backslash e)) = \{\{\langle -2,1\rangle,\langle 1,0\rangle\}\},\\
&\#(\text{antecedent}) = \{\{\langle -1,1\rangle,\langle 2,0\rangle,\langle -2,1\rangle,\langle 1,0\rangle\},\{\langle 0,1\rangle,\langle 3,0\rangle,\langle -2,1\rangle,\langle 1,0\rangle\}\},\\
&\#(t\cap^{+}e) = \{\{\langle 0,1\rangle\},\{\langle 1,0\rangle\}\}.
\end{aligned}$$

This generalized count invariance is not a 'symmetric' notion; it correctly accepts $a\cap b \Rightarrow a$ while rejecting $a \Rightarrow a\cap b$. Of course, it is still a very rough measure and does not capture the logic of $\cap$ correctly.

Example. The generalized count check accepts both of the following sequents, while only the former is derivable in $\mathsf{L}(/,\backslash,\bullet,\cap)$:

$$((a\cap^{-} b)\backslash c)\backslash d \Rightarrow ((a\backslash c)\backslash d)\cap^{+}((b\backslash c)\backslash d)$$
$$((a\backslash c)\backslash d)\cap^{-}((b\backslash c)\backslash d) \Rightarrow ((a\cap^{+} b)\backslash c)\backslash d$$

Here is a numerical countermodel for the latter sequent (O and E are the sets of odd and even natural numbers, respectively):

$$\begin{array}{ccccccccccccc}
((a & \backslash & c\,) & \backslash & d\,) & \cap & ((b & \backslash & c\,) & \backslash & d\,) \\
\mathsf{O} & & \mathsf{E}-\{0\} & & \mathsf{N}-\{0\} & & \mathsf{E} & & \mathsf{E}-\{0\} & & \mathsf{N}-\{0\} \\
 & \mathsf{O} & & & & & & \mathsf{E}-\{0\} & & & \\
 & & & \mathsf{N} & & & & & & \mathsf{N} & \\
 & & & & & \mathsf{N} & & & & &
\end{array}$$

$$\begin{array}{cccccccc}
\Rightarrow & ((a & \cap & b\,) & \backslash & c\,) & \backslash & d \\
 & \mathsf{O} & & \mathsf{E} & & \mathsf{E}-\{0\} & & \mathsf{N}-\{0\} \\
 & & \emptyset & & & & & \\
 & & & & \mathsf{N} & & & \\
 & & & & & & \mathsf{N}-\{0\} &
\end{array}$$

§6. Adding disjunction

Finally, we consider the Lambek calculus with an added disjunction. Our numerical models and generalized type count both admit of natural extension to cover disjunction. In numerical models, disjunction is naturally interpreted as union. In count check, $a\cup^{+} b$ and $a\cup^{-} b$ can be given the same value as $a\cap^{-} b$ and $a\cap^{+} b$, respectively. We leave the verification to the reader.

We prove analogues of Theorems I–IIb about disjunction.

THEOREM VIII. *$\mathsf{L}(/,\backslash,\bullet,\cap,\cup)$ recognizes any finite union of $\mathsf{L}(/,\backslash,\bullet,\cap)$-recognizable languages. (Similarly with calculi with a smaller set of connectives.)*

Proof. Let L_1 and L_2 be recognized by $\mathsf{L}(/,\backslash,\bullet,\cap)$-grammars G_1 and G_2 with distinguished types t_1 and t_2, respectively, where the types used in G_1 and G_2 are distinct. We construct a $\mathsf{L}(/,\backslash,\bullet,\cap,\cup)$-grammar $G_1\cup G_2$ that recognizes $L_1\cup L_2$ as follows. Let $G_1\cup G_2$ retain all type assignments from G_1 and G_2. Let the distinguished type of $G_1\cup G_2$ be $t_1\cup t_2$. What we need to show is the following:

Suppose that X consists of pure G_1-types and/or pure G_2-types. Then

$$X \Rightarrow t_1 \cup t_2$$

is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap, \cup)$ if and only if either

$$X \Rightarrow t_1$$

or

$$X \Rightarrow t_2$$

is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap)$.

One direction is obvious. For the other direction, we can use our permutability argument again, to show any cut-free derivation of $X \Rightarrow t_1 \cup t_2$ can be transformed to one in which $(\Rightarrow \cup)$ is the last rule applied, noting that there is no application of $(\cup \Rightarrow)$. This can be checked easily. (Alternatively, we can use the method used in the proof of Theorem VII.)

Finally, note that if $X \Rightarrow t_i$ is derivable, X must consist entirely of pure G_i-types, for $i = 1, 2$.

THEOREM IX. *The* $\mathsf{L}(/, \backslash, \bullet, \cap, \cup)$*-recognizable languages are closed under union. (Similarly with other calculi with a smaller set of connectives having* $\cup$*.)*

Proof. An argument similar to the one in the proof of Theorem II can be used, but let us take the method used in the proof of Theorem VII here. Let A and B be disjoint sets of basic types. Let $X \Rightarrow t_1 \cup t_2$ be a sequent in $\mathsf{L}(/, \backslash, \bullet, \cap, \cup)$ such that t_1 is a pure A-type, t_2 is a pure B-type, and each type in X is either a pure A-type or a pure B-type. We prove that if $X \Rightarrow t_1 \cup t_2$ is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap, \cup)$, then either $X \Rightarrow t_1$ or $X \Rightarrow t_2$ is derivable in $\mathsf{L}(/, \backslash, \bullet, \cap, \cup)$.

Suppose $X \Rightarrow t_1 \cup t_2$ to be derivable. First, we note that X must consist entirely of pure A-types or of pure B-types, using numerical models. Recall that $\cup$ is naturally interpreted as union. Suppose the antecedent X is a mixture of pure A-types and pure B-types. Then assigning $\mathsf{N} \times \{0\}$ to all basic types in A and $\{0\} \times \mathsf{N}$ to all basic types in B constitutes a numerical countermodel to $X \Rightarrow t_1 \cup t_2$.

Suppose X consists entirely of pure A-types. We take a cut-free derivation of $X \Rightarrow t_1 \cup t_2$ and transform it to a cut-free derivation of $X \Rightarrow t_1$. We do this by replacing all occurrences of $t_1 \cup t_2$ in the given derivation by t_1. Each rule application where $t_1 \cup t_2$ is a parametric formula or a side formula remains an application of the same rule after the replacement. The only fault of the resulting 'derivation' is that what used to be applications of $(\Rightarrow \cup)$ with $t_1 \cup t_2$ as principal formula are not instances of inference any more (note that $t_1 \cup t_2$ cannot originate in an axiom):

$$\frac{X' \Rightarrow t_i}{X' \Rightarrow t_1 \cup t_2} \rightsquigarrow \frac{X' \Rightarrow t_i}{X' \Rightarrow t_1}$$

$i = 1$ or 2. Note that there can be many situations like this, because of applications of $(\cup \Rightarrow)$, which can contract occurrences of $t_1 \cup t_2$. Here, i must be 1 in each such situation, since X' must consist entirely of pure A-types. Then these steps can be eliminated, just deleting the lower or higher occurrence of $X' \Rightarrow t_1$. We now have a derivation of $X \Rightarrow t_1$.

The implications of Theorems VIII and IX are less clear in this case, however, since the context-free languages are already closed under union. One can also show that the languages recognized by $\mathsf{L}(/, \backslash)$ via a *basic* distinguished type are closed under union, by the method used for LP in [2]. This particular method does not work for grammars with complex distinguished types, so we do not know whether the $\mathsf{L}(/, \backslash)$-recognizable languages are closed under union.[9]

Question. Are all $\mathsf{L}(/, \backslash)$-recognizable languages recognizable via a $\mathsf{L}(/, \backslash)$-grammar with a basic distinguished type?[10]

Also, we do not know whether addition of $\cup$ does not have any other effect on the recognizing power, as in the case of $\cap$.

Question. Are $\mathsf{L}(/, \ldots, \cup)$-recognizable languages already $\mathsf{L}(/, \ldots)$-recognizable?

§7. Linguistic utility of intersective conjunction and disjunction

Intersective conjunction and disjunction might be of some use for linguistic purposes. Besides its potential for achieving single type assignment, intersective conjunction may also be used to encode feature decomposition of categories. Disjunction can naturally be used for disjunctive specification of arguments. For example, the following type assignment may be used for a (simplified) grammar of English.

walks	$\longmapsto$	$(np \cap sing)\backslash s$
walk	$\longmapsto$	$(np \cap pl)\backslash s$
walked	$\longmapsto$	$np\backslash s$
John	$\longmapsto$	$np \cap sing$
the Beatles	$\longmapsto$	$np \cap pl$
the Chinese	$\longmapsto$	$np \cap (sing \cap pl)$
became	$\longmapsto$	$(np\backslash s)/(np \cup ap)$
famous	$\longmapsto$	ap

The grammar recognizes such strings as **John walks**, **The Beatles walked**, **The Chinese walk**, **The Beatles became famous**, **The Chinese became the Beatles**, etc., but not **John walk** or **The Beatles walks**. Note the type assigned to **the Chinese**. Since the expression can be used both as a singular noun phrase and as a plural noun phrase, the type must be the conjunction of np, $sing$, and pl. Contrary to what is done in unification-based grammars, assigning np or $np \cap (sing \cup pl)$ to it will not do.

Our result about the recognizing effect of intersective conjunction warns against unconstrained use of it in categorial grammars for natural languages. The recognizing power of Lambek calculi with intersective conjunction seems too strong. As we have seen, the recognizable languages include some languages which lack the property of semi-linearity (e.g., $\{(a^n b^n)^n \mid n \geq 1\}$ used in Lemma 2 (p. 11)).

However, in linguistic application, it seems that the full apparatus of $\mathsf{L}(\ldots, \cap)$ need not be used in a way which leads to such a strong recognizing power. The right introduction rule $(\Rightarrow \cap)$ does not seem to be essential in a grammar like the one seen above. For example, consider the following sequents, which license **John walks**, **The Beatles walked**, and

[9] In fact, exactly the same thing can be said about the effect of adding $\bullet$ to Lambek calculi without it. With $\bullet$, a grammar recognizing the concatenation of the two $\mathsf{L}(/, \backslash)$-recognizable languages can be constructed easily, while the method used in [2] shows that one can get by without $\bullet$ if the original languages are recognized via basic distinguished types.

[10] See the first remark in the section on single type assignment.

The Chinese walk:

$$np \cap sing,\ (np \cap sing)\backslash s \ \Rightarrow\ s$$
$$np \cap pl,\ np\backslash s \ \Rightarrow\ s$$
$$np \cap (sing \cap pl),\ (np \cap pl)\backslash s \ \Rightarrow\ s$$

The first two are derivable using only $(\cap \Rightarrow)$ (as well as axioms and rules for $\backslash$). The third sequent is not derivable without using $(\Rightarrow \cap)$:

$$\dfrac{\dfrac{\dfrac{np \Rightarrow np}{np \cap (sing \cap pl) \Rightarrow np} \quad \dfrac{\dfrac{pl \Rightarrow pl}{sing \cap pl \Rightarrow pl}}{np \cap (sing \cap pl) \Rightarrow pl}}{np \cap (sing \cap pl) \Rightarrow np \cap pl} \quad s \Rightarrow s}{np \cap (sing \cap pl), (np \cap pl)\backslash s \Rightarrow s}$$

However, the use of $(\Rightarrow \cap)$ can be avoided by slightly changing the type assigned to the Chinese, namely to $(np \cap sing) \cap (np \cap pl)$. In general, the following seems to be the case: given a $\mathsf{L}(\ldots, \cap)$-grammar of the kind we are considering, an equivalent $\mathsf{L}(\ldots, \cap^-)$-grammar can be obtained by rebracketing and rearranging the conjuncts of conjunctive types used in the original grammar. Then the recognizing power of such $\mathsf{L}(\ldots, \cap)$-grammars does not exceed that of $\mathsf{L}(\ldots, \cap^-)$.

Similarly, as for disjunction, use of $(\cup \Rightarrow)$ does not seem to be essential for the kind of linguistic application we are considering. Our example sentences The Beatles became famous and The Chinese became the Beatles are recognized by the above grammar without using $(\cup \Rightarrow)$. This is significant, because in the presence of $(\cap \Rightarrow)$, $(\cup \Rightarrow)$ is as 'dangerous' as $(\Rightarrow \cap)$. If L is a finite intersection of $\mathsf{L}(/, \backslash)$-recognizable languages, then $\mathbf{z}L$, where $\mathbf{z}$ is a new symbol, is $\mathsf{L}(/, \backslash, \cap^-, \cup^-)$-recognizable.[11]

We can show that the presence of $\cap$ and $\cup$ does not lead to increased recognizing power if $(\Rightarrow \cap)$ and $(\cup \Rightarrow)$ are not used:

THEOREM X. *If L is recognized by $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$ via a basic distinguished type, then L is $\mathsf{L}(\overline{C})$-recognizable (also via a basic type), where $\overline{C}$ is a selection from $\{/, \backslash, \bullet\}$.*

Proof. Let us call a subtype occurrence b in a $\cap$-$\cup$-*maximal in* a if $b = c \cap d$ or $c \cup d$ and b does not lie within the scope of $\cap$ or $\cup$. A positive conjunctive $\cap$-$\cup$-maximal subtype occurrence, etc., are understood in the obvious sense. For any type a, let $U(a)$ be the set of types which can be obtained from a by replacing a positive conjunctive $\cap$-$\cup$-maximal subtype occurrence $c \cap d$ in it by c or by d, or by replacing a negative disjunctive $\cap$-$\cup$-maximal subtype occurrence $c \cup d$ in it by c or by d. By applying this operation U to members of $U(a)$, and so on, we get sets $U^n(a)$ for each $n \geq 0$. Let $U^*(a) = \bigcup_{n \geq 0} U^n(a)$. $U^*(a)$ is finite. By straightforward induction, if $a' \in U(a)$, then $a \Rightarrow a'$ is derivable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$. By Cut, then, if $a' \in U^*(a)$, $a \Rightarrow a'$ is derivable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$. Again by Cut, if $a'_1, \ldots, a'_n \Rightarrow t$ is derivable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$ and each a'_i is in $U^*(a_i)$ $(1 \leq i \leq n)$, then $a_1, \ldots, a_n \Rightarrow t$ is also derivable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$. Moreover, it can be shown that, if $a_1, \ldots, a_n \Rightarrow t$ is derivable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$ and t is a basic type, then there is some $a'_i \in U^*(a_i)$ for each i such that $a'_1, \ldots, a'_n \Rightarrow t$ is derivable in $\mathsf{L}(\overline{C})$ (regarding subtypes of the form $c \cap d$ or $c \cup d$ as basic types). To see this, take a cut-free derivation $\mathcal{F}$ of

[11] Proof is by a method similar to that of Theorem I. $\mathbf{z}$ is assigned a disjunctive type of the form $t/t_1 \cup \cdots \cup t/t_n$, where t is a new distinguished type and t_i's are the distinguished types of the given $\mathsf{L}(/, \backslash)$-grammars.

$a_1, \ldots, a_n \Rightarrow t$ in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$. (Recall that Cut is eliminable in $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$.) We 'skip' all applications of $(\cap \Rightarrow)$ and $(\Rightarrow \cup)$ in $\mathcal{F}$, starting from the lowest ones and working up along each branch. The result is a new derivation, for, in the absence of $(\Rightarrow \cap)$ and $(\cup \Rightarrow)$, no rule application can be made impossible by skipping $(\cap \Rightarrow)$ and $(\cup \Rightarrow)$. (The detail is left to the reader. The method is similar to that used in the proof of Theorem VII.) At each step, skipping a lowest application of $(\cap \Rightarrow)$ or $(\Rightarrow \cup)$ results in replacing an antecedent formula a in the endsequent by some a' in $U(a)$. (Note that positive conjunctive $\cap$-$\cup$-maximal subtype occurrences in a occur negatively in the sequent, and negative disjunctive $\cap$-$\cup$-maximal subtype occurrences in a occur positively in the sequent.) Some occurrences of subtypes of the form $c \cap d$ or $c \cup d$ may remain, but they originate in axioms. The result is a cut-free $\mathsf{L}(\overline{C})$-derivation of $a'_1, \ldots, a'_n \Rightarrow t$, for some $a'_i \in U^*(a_i)$ $(1 \leq i \leq n)$.

Let G be a $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$-grammar that recognizes L. Construct a grammar G' as follows. If G assigns type a to symbol s, let G' assign all types in $U^*(a)$ to s. From the foregoing analysis, it is clear that L is recognized by $\mathsf{L}(\overline{C})$ via G' (regarding subtypes of the form $c \cap d$ or $c \cup d$ used in G' as basic types).

'Basic distinguished type' in Theorem X can be replaced by 'distinguished type which does not contain negative conjunctive $\cap$-$\cup$-maximal subtype occurrences or positive disjunctive $\cap$-$\cup$-maximal subtype occurrences'. For the general case, the following holds:

THEOREM XI. *If L is $\mathsf{L}(\overline{C}, \cap^-, \cup^+)$-recognizable, then L is a finite union of $\mathsf{L}(\overline{C})$-recognizable languages, where $\overline{C}$ is a selection from $\{/, \backslash, \bullet\}$.*

Proof. This time, the distinguished type t of the original grammar will be replaced by elements of $V^*(t)$, the set of types which can be obtained from t by repeatedly replacing negative conjunctive $\cap$-$\cup$-maximal subtype occurrences $c \cap d$ by c or by d and positive disjunctive $\cap$-$\cup$-maximal subtype occurrences $c \cup d$ by c or by d. For each t' in $V^*(t)$, construct a grammar with t' as its distinguished type, in the way G' is constructed in the proof of Theorem X.

§8. Prospects

In this paper, we have studied certain logically natural extensions of the Lambek calculus, which have analogues of conjunction and disjunction. Other kinds of connectives can also be considered in the Lambek calculus. For example, it would be of interest to study the Lambek calculus counterparts of all the connectives used in Linear Logic, including modalities.

Ultimately, the specific results obtained about individual connectives will call for a more general, abstract characterization. For instance, what are the general characteristics of connectives which, when added to the original Lambek calculus, increase the recognizing power of the latter?[12] Can we always 'read off' from the rules for a connective the closure property of the recognizable family of languages it ensures?

Some of the points made in the present paper may inspire an even more abstract kind of question. Is it possible to give a less trivial 'definability' result than Lemma 1 (p. 11)? That is, can we find an interesting common property of families of languages that are definable by some system of sequent calculus (with or without structural rules) via categorial grammars?

[12]The results in §7 might suggest that introduction rules which contract parametric formulas bring in something new.

Further study of categorial grammars and their logics will surely raise many other fundamental questions in the interface of logic and mathematical linguistics.[13]

References

[1] Benthem, J. van, Semantic type change and syntactic recognition, in G. Chierchia, B. H. Partee, and R. Turner (eds.), *Properties, Types and Meaning, Volume I: Foundational Issues*, Kluwer, Dordrecht, 1989.

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

[3] Buszkowski, W., Compatibility of a categorial grammar with an associated category system, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* **28** (1982), 229–238.

[4] 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.

[5] Buszkowski, W., Generative power of categorial grammars, in R. Oehrle, E. Bach, and D. Wheeler (eds.), *Categorial Grammars and Natural Language Structures*, Reidel, Dordrecht, 1988.

[6] Buszkowski, W., On generative capacity of the Lambek calculus, in J. van Eijk (ed.), *Logics in AI, Lecture Notes in Artificial Intelligence 478*, Springer, Berlin, 1991.

[7] Curry, H. B., The permutability of rules in the classical inferential calculus, *Journal of Symbolic Logic* **17** (1952), 245–248.

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

[9] Kleene, S. C., Permutability of inferences in Gentzen's calculi *LK* and *LJ*, in *Two Papers on the Predicate Calculus, Memoirs of the American Mathematical Society* **10**, 1952.

[10] Lambek, J., The mathematics of sentence structure, *American Mathematical Monthly* **65** (1958), 154–170. (Also in W. Buszkowski, W. Marciszewski, and J. van Benthem (eds.), *Categorial Grammars*, John Benjamins, Amsterdam, 1988.)

[11] Lambek, J., On the calculus of syntactic types, in R. Jakobson (ed.), *Structure of Language and its Mathematical Aspects*, American Mathematical Society, 1961.

[12] Păun, G., A note on the intersection of context-free languages, *Fundamenta Informaticae* **3** (1980), 135–139.

[13] Van Benthem [2] asks whether there is some standard proof-theoretic relation which coincides with the relation of relative strength among categorial calculi with respect to their recognizing power.

The ITLI 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

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

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

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 Diagonizable 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