Institute for Logic, Language and Computation

PROVING THEOREMS OF THE LAMBEK CALCULUS OF ORDER 2 IN POLYNOMIAL TIME

Erik Aarts

ILLC Prepublication Series
for Computation and Complexity Theory CT-93-05

University of Amsterdam

The ILLC Prepublication Series

1990

Logic, Semantics and Philosophy of Language

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

Mathematical Logic and Foundations

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

Computation and Complexity Theory

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

Other Prepublications

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

1991

Logic, Semantics and Philosophy of Langauge

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

Mathematical Logic and Foundations

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

Computation and Complexity Theory

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

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

PROVING THEOREMS OF THE LAMBEK CALCULUS OF ORDER 2 IN POLYNOMIAL TIME

Erik Aarts
Department of Mathematics and Computer Science
University of Amsterdam
OTS, University of Utrecht

ILLC Prepublications
for Computation and Complexity Theory
ISSN 0928-3323

Coordinating editor: Dick de Jongh

Research supported by
the Netherlands Organization for Scientific Research (NWO)

received May 1993

Proving theorems of the Lambek calculus of order 2 in polynomial time

Erik Aarts*

OTS
Utrecht University
Trans 10
3512 JK Utrecht

Department of Mathematics and Computer Science
University of Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

Abstract

In the Lambek calculus of order 2 we allow only sequents in which the depth of nesting of implications is limited to 2. We prove that the decision problem of provability in the calculus can be solved in time polynomial in the length of the sequent. A normal form for proofs of second order sequents is defined. It is shown that for every proof there is a normal form proof with the same reading. With this normal form we can give two algorithms that decide about provability of sequents in polynomial time.

1 Introduction

The Lambek calculus was first presented in Lambek [1958]. The Lambek calculus is a sequent calculus. A sequent expresses a consequence relation between a list of types (the antecedent) and a succedent type. Types are defined recursively:

- There is a set of basic types (e.g. {s, np}).
- If A and B are types, then A/B and A\B are types.

We consider the (product-free) Lambek calculus, of which we give a Gentzen-style sequent presentation in figure 1. Greek capitals denote sequences of types. Empty antecedents are not allowed.
In the left rules, we call $\Gamma \vdash \mathsf{A}$ the *minor* premise and $\Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}$ the *major* premise.

*The author was sponsored by project NF 102/62-356 ('Structural and Semantic Parallels in Natural Languages and Programming Languages'), funded by the Netherlands Organization for the Advancement of Research (NWO).

$$\mathsf{A} \vdash \mathsf{A} \quad (\mathsf{A}\ \textit{basic})$$

$$\frac{\Gamma \vdash \mathsf{A} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, \mathsf{B}/\mathsf{A}, \Gamma, \Delta' \vdash \mathsf{C}}\ [/\mathrm{L}] \qquad \frac{\Gamma, \mathsf{A} \vdash \mathsf{B}}{\Gamma \vdash \mathsf{B}/\mathsf{A}}\ [/\mathrm{R}]$$

$$\frac{\Gamma \vdash \mathsf{A} \quad \Delta, \mathsf{B}, \Delta' \vdash \mathsf{C}}{\Delta, \Gamma, \mathsf{A}\backslash\mathsf{B}, \Delta' \vdash \mathsf{C}}\ [\backslash\mathrm{L}] \qquad \frac{\mathsf{A}, \Gamma \vdash \mathsf{B}}{\Gamma \vdash \mathsf{A}\backslash\mathsf{B}}\ [\backslash\mathrm{R}]$$

Figure 1: The Lambek calculus

Moortgat [1988] already showed that Lambek proofs are small, their size is linear in the number of connectives in a sequent. We can conclude from this that provability in the Lambek calculus is in the complexity class $\mathcal{NP}$. All algorithms for theorem proving that have been developed ([Moortgat, 1988], [Hendriks, 1992]) run in time exponential in the size of the sequent that must be proved. There are no polynomial algorithms known. In this paper a fragment of the Lambek calculus is considered for which polynomial algorithms do exist, and the algorithms are presented here. The fragment is called *the second order fragment.*

Buszkowski [1990] proved contextfreeness of the second order fragment. There is no obvious way to use the contextfreeness in finding polynomial algorithms for the fragment. Buszkowski says in his proof [Buszkowski, 1990, p. 149]: "Details of the construction, however, are pretty sophisticated (it would be interesting to estimate the required time which seems to be super-exponential)". The grammar he gives is contextfree but exponential in size. Via Buszkowski's construction it is not possible to obtain a polynomial algorithm for theorem proving in the second order fragment.

After description of two polynomial algorithms it is shown that, unfortunately, the method used in this paper can not be extended to larger fragments.

The fragment can be defined as follows. We define the *order of a type*:

$$\begin{aligned} order(A) &= 0 \text{ if A is a basic type} \\ order(A/B) &= max(order(A), order(B)+1) \\ order(B\backslash A) &= max(order(A), order(B)+1) \end{aligned}$$

Maybe it would be better to call this property of a type its depth but we call it order because others did so before. The *order of a sequent* is defined as the order of the highest order type occurring in it.

The fragment we restrict ourselves to in this paper is the fragment of sequents of order 0, 1 or 2. To give an example: the sequent $s/(np\backslash s), np\backslash(s/np), (s/np)\backslash s \vdash s$ has order 2. The sequent $s/(s/(np\backslash s)), np \vdash s$ has order 3 and is not in the fragment. For convenience we only look at sequents whose succedent is a basic type[1].

[1]We can easily transform sequents with a complex succedent to sequents with a basic succedent by application of right rules.

2 A normal form for proofs

Before we give the normal form we change the notation of types. Because the system is associative the bracketing of a type is irrelevant to some extent. We remove this unnecessary bracketing by giving a type two *subcategorization lists*: one for the arguments that are expected at the left and one for the arguments that are expected at the right of some type. A type $((a\backslash(b\backslash c))/d)/e$ is converted into the type in subcategorization list notation `(c,[a,b],[e,d])`. Note that the subcategorization lists can be nested. We can define the Lambek calculus in this new notation as follows (we use the notation `[H|T]` for the list with head H and tail T and `(a,[],[])` equals a):

$$\mathsf{A} \vdash \mathsf{A} \quad (\mathsf{A}\ \textit{basic})$$

$$\frac{\Gamma \vdash \mathsf{A} \quad \Delta, (\mathsf{B}, \mathsf{L}, \mathsf{T}), \Delta' \vdash \mathsf{C}}{\Delta, (\mathsf{B}, \mathsf{L}, [\mathsf{A}|\mathsf{T}]), \Gamma, \Delta' \vdash \mathsf{C}}\,[/\mathrm{L}] \qquad \frac{\Gamma, \mathsf{A} \vdash (\mathsf{B}, \mathsf{L}, \mathsf{T})}{\Gamma \vdash (\mathsf{B}, \mathsf{L}, [\mathsf{A}|\mathsf{T}])}\,[/\mathrm{R}]$$

$$\frac{\Gamma \vdash \mathsf{A} \quad \Delta, (\mathsf{B}, \mathsf{T}, \mathsf{L}), \Delta' \vdash \mathsf{C}}{\Delta, \Gamma, (\mathsf{B}, [\mathsf{A}|\mathsf{T}], \mathsf{L}), \Delta' \vdash \mathsf{C}}\,[\backslash\mathrm{L}] \qquad \frac{\mathsf{A}, \Gamma \vdash (\mathsf{B}, \mathsf{L}, \mathsf{T})}{\Gamma \vdash (\mathsf{B}, [\mathsf{A}|\mathsf{T}], \mathsf{L})}\,[\backslash\mathrm{R}]$$

Figure 2: The Lambek calculus in subcategorization list notation

The system is equivalent with the Lambek calculus because the Lambek calculus is associative. The definition of order in this system is analog with the definition we just gave.
In this new notation we can define a normal form as follows:

- the sub-sequents in the proof must lie on three intervals on the unfolded proof frame as defined in [Roorda, 1991].

Roorda defines a mapping from types to trees. This mapping is called unfolding. In order to unfold types we first assign the labels "+" and "−" to them. Types in the antecedent are labeled negative, in the succedent positive.
We first define an initial tree for a type. This tree has only one node. It is the type with its label. There are four rules to unfold nodes:

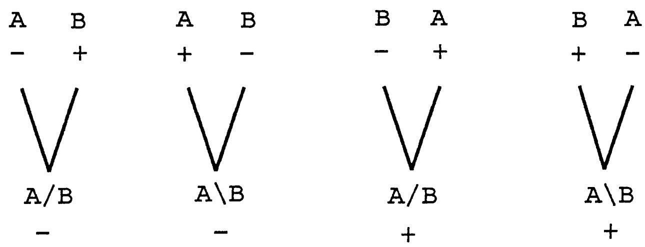


Figure 3: Unfolding rules

If all leaves of the tree are unfolded the unfolding is finished. If we unfold all types of a sequent, we get the *proof frame* of a sequent.
E.g. the proof frame of the sequent: $np, np\backslash(s/np) \vdash s/np$ is:

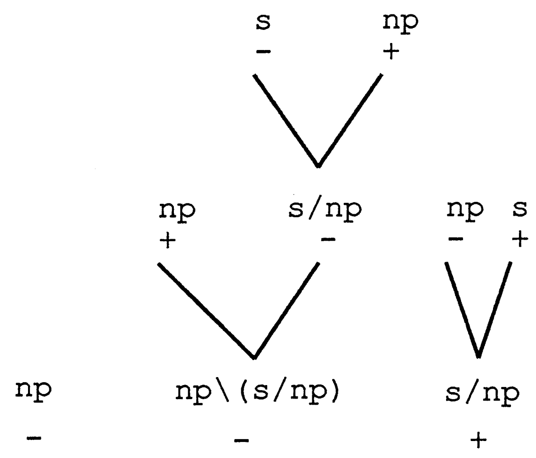


Figure 4: A proof frame

A proof frame is a list of trees. We define an *interval* as a list of leaves that are adjacent in the proof frame. We define the last leaf of the last tree to be adjacent with the first leaf of the first tree. This means that a proof frame is circular instead of linear. We will draw the proof frames circular in the rest of the paper.
We call the final conclusion in a proof the goal sequent. We can give proof frames for all sub-sequents in a cut free (Gentzen) proof in the same way as for the goal sequent. Because the subformula property holds in the cut free system, the leaves in a sub proof frame form a subset of the leaves in the proof frame of the goal sequent.
We can count the number of intervals the leaves of a sub-sequent are on with respect to the leaves of the proof frame of the goal sequent. Consider e.g. the goal sequent $a/b, c/d, e/f, g, h\backslash i, j\backslash k, l\backslash m \vdash n$. The sub-sequent $c/d, g, h\backslash i \vdash n$ is on three intervals with respect to the goal sequent. The sub-sequent $c/d, g, j\backslash k \vdash n$ is on four intervals. Figure 5 will make this clear.
The normal form for proofs is now defined as follows: *all sub-sequents in the proof must lie on at most three intervals.* Therefore, a proof of the sequent $a/b, c/d, e/f, g, h\backslash i, j\backslash k, l\backslash m \vdash n$ containing the premise $c/d, g, j\backslash k \vdash n$ is not in normal form.
Definition of this normal form reduces the number of possible sub-sequents occurring in a proof enormously. There are exponentially many possible sub-sequents of a sequent (the set of leaves has exponentially many subsets). We restrict the number of possible sub-sequents (subsets) by admitting only sub sequents whose leaves form at most three intervals.

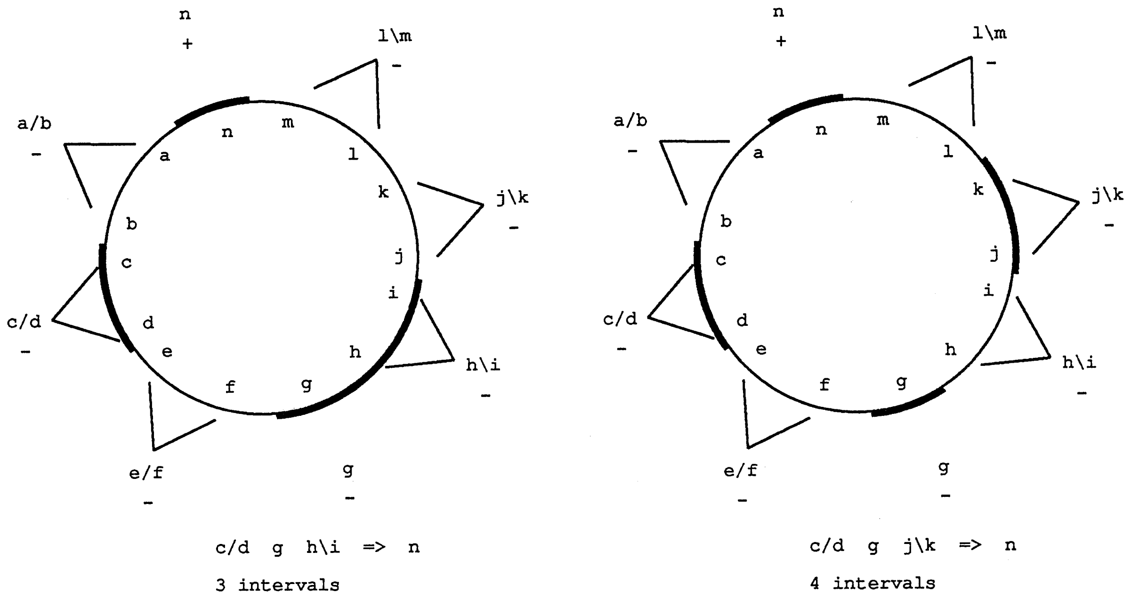


Figure 5: Three vs. four intervals

When there are n leaves in the proof frame, there are fewer than $\binom{n}{6}$ possibilities to choose three intervals (draw 6 borders out of n possible ones). The number $\binom{n}{6}$ is smaller than n^6. If we have chosen three intervals, there is a (possibly empty) set of sequents whose leaves are on these three intervals. Only one of them (if there are any) has a basic succedent. All others can be derived from this one by application of right rules only.
Under the 3-intervals restriction the number of possible sub-sequents (subsets) is polynomial ($\mathcal{O}(n^6)$ if we consider sequents with a basic succedent).

Suppose we have a proof that is not in normal form. Then we have to show that there is an equivalent proof (a proof with the same axioms) which is in normal form.

The basic idea is as follows. We take as an induction hypothesis that we try to prove sequents whose antecedent types are on one interval. The succedent type is always on one interval so the sequent is on (at most) two intervals. The induction hypothesis is clearly true for the goal sequent.
A standard normal form for proofs is downward permutation of right rules. We bring the proof in this normal form first. Reasoning from the bottom to the top of a proof, right rules are applied first. If no further right rules are applicable, the antecedent has the following form: first we have a number of "fresh" adjacent types that have just been moved from the succedent to the antecedent via the [\R] right rule. They are followed by the old antecedent which is followed by the "fresh" types that are a result of the [/R] rule. The

antecedent type involved in a [\R] or [/R] rule is called hypothetic because the type would be a hypothesis in a natural deduction proof of the sequent. The hypothetic "fresh" types are always basic because we are in the second order fragment.

If no right rules are applicable from the start *(case 1)* we can bring the proof in head normal form [Hendriks, 1992]. The basic idea is that if we apply a left rule on a type in the conclusion the type becomes "active". In the next step a left rule must be applied on the functor of the active type in the major premise. If we enrich the language with the symbol $*$ we can define a new calculus in which all proofs are in head normal form as follows:

$$at* \vdash at \quad (at \ is \ basic)$$

$$\frac{\Gamma \vdash \mathsf{A}* \quad \Delta, \mathsf{B}*, \Delta' \vdash at}{\Delta, \mathsf{B}/\mathsf{A}*, \Gamma, \Delta' \vdash at}\ [/\mathrm{L}] \qquad \frac{\Gamma, \mathsf{A} \vdash \mathsf{B}*}{\Gamma \vdash \mathsf{B}/\mathsf{A}*}\ [/\mathrm{R}]$$

$$\frac{\Gamma \vdash \mathsf{A}* \quad \Delta, \mathsf{B}*, \Delta' \vdash at}{\Delta, \Gamma, \mathsf{A}\backslash\mathsf{B}*, \Delta' \vdash at}\ [\backslash\mathrm{L}] \qquad \frac{\mathsf{A}, \Gamma \vdash \mathsf{B}*}{\Gamma \vdash \mathsf{A}\backslash\mathsf{B}*}\ [\backslash\mathrm{R}]$$

$$\frac{\Delta, \mathsf{A}*, \Delta' \vdash at}{\Delta, \mathsf{A}, \Delta' \vdash at*}\ [*]$$

Figure 6: The Lambek $*$ calculus

We can apply this idea on the proofs we have in our system. After this normalization step we normalize further: we separate the [\L] and [/L] rules by applying all [\L] first and all [/L] rules afterwards. This is possible because we use the subcategorization list calculus. The antecedents of the minor premises are on one interval. This is the case because the antecedent of a minor premise is a sublist of the antecedent of the conclusion. In head normal form every antecedent of a minor premise is a sublist of the final conclusion. The antecedent of the final conclusion was on one interval (induction hypothesis) and so are the antecedents of the minor premises. The antecedents of the major premises are on two intervals. Because of the head normal form and separation of [\L] and [/L] rules the antecedent of the final conclusion is split up in at most two parts. The last major premise is an axiom. Axioms are on two intervals. This means that the induction hypothesis is valid for all sub-sequents we have not proved yet.

$$\frac{minor-2 \qquad \dfrac{minor-2 \qquad \dfrac{\dfrac{minor-2 \qquad \dfrac{minor-2 \quad axiom-2}{\vdots\ ([/\mathrm{L}])}}{}\ \ major-3}{major-3}\ [/\mathrm{L}]}{\vdots\ ([\backslash\mathrm{L}])}\ \ major-3}{conclusion-2}\ [\backslash\mathrm{L}]$$

Suppose right rules were applicable *(case 2)* and that there is at least one [\R] rule. Then the proof is of the form:

$$
\dfrac{\mathsf{B} \vdash \mathsf{B} \quad \vdots \; \Gamma_5, (\mathsf{A}, \Delta_2, \Delta_4), \Gamma_6 \vdash \mathsf{D}}{\Gamma_5, \mathsf{B}, (\mathsf{A}, [\mathsf{B}, \Delta_2], \Delta_4), \Gamma_6 \vdash \mathsf{D}}\,[\backslash L]
$$
$$
\vdots\; \Pi_2([\backslash L], [/L])
$$
$$
\Gamma_1, \mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}
$$
$$
\vdots\; \Pi_1([\backslash R], [/R])
$$
$$
\Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3 \vdash (\mathsf{C}, [\mathsf{B}, \Gamma_1^R], \Gamma_4)
$$

The induction hypothesis says that $\Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3$ is on one interval. The type that has the B as an argument can not be a member of Γ_4 because Γ_4 is a list of basic types. If we have such a proof, we can bring it in the following form:

$$
\vdots
$$
$$
\Gamma_5, (\mathsf{A}, \Delta_2, \Delta_4), \Gamma_6 \vdash \mathsf{D}
$$
$$
\vdots
$$
$$
\dfrac{\mathsf{B} \vdash \mathsf{B} \quad \Gamma_1, (\mathsf{A}, \Delta_2, \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}}{\Gamma_1, \mathsf{B}, (\mathsf{A}, [\mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}}\,[\backslash L]
$$
$$
\vdots\; \Pi_3([\backslash L])
$$
$$
\Gamma_1, \mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}
$$
$$
\vdots\; \Pi_1([\backslash R], [/R])
$$
$$
\Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3 \vdash (\mathsf{C}, [\mathsf{B}, \Gamma_1^R], \Gamma_4)
$$

The idea is that when there are hypothetic types in the antecedent we try to remove them as soon as possible so that the sequent that still has to proved is an instance of case 1 we just treated. In terms of permutation of rules we try to bring the axiom $\mathsf{B} \vdash \mathsf{B}$ as far down as possible. We adopt some kind of restricted head normal form: in the partial proof Π_3 (and only in Π_3) the type $(\mathsf{A}, [\ldots, \mathsf{B}, \Delta_2], \Delta_3)$ is active (has a $*$) temporarily. When the B has been consumed the type "looses its activity".

The proof can be brought in this form because the rules on the partial antecedent $\mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3)$ are independent from all other rules.

Note that although the antecedent $\Gamma_1, \mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4$ is on three intervals, the whole sequent $\Gamma_1, \mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}$ is on two intervals because Γ_4, C and Γ_1 are adjacent. In the partial proof Π_3 only [\L] rules are applied on $(\mathsf{A}, [\ldots, \mathsf{B}, \Delta_2], \Delta_3)$. Γ_2 is on one interval by the induction hypothesis. The antecedents of the minor premises in Π_3 are sublists of Γ_2 and therefore are on one interval too.

In the major premises in Π_3 the antecedent is broken up in two parts compared to the antecedent $\Gamma_1, \mathsf{B}, \Gamma_2, (\mathsf{A}, [\Delta_1, \mathsf{B}, \Delta_2], \Delta_3), \Gamma_3, \Gamma_4$. Therefore the ma-

jor premises lie on three intervals. In the sequent $\Gamma_1, (\mathsf{A}, \Delta_2, \Delta_3), \Gamma_3, \Gamma_4 \vdash \mathsf{C}$ the part $\Gamma_4, \mathsf{C}, \Gamma_1$ is on one interval and $(\mathsf{A}, \Delta_2, \Delta_3), \Gamma_3$ is on one interval.
We can repeat this story for all basic hypothetic types in Γ_1. We remove the hypotheses one by one. Then we eliminate the hypotheses that are moved from the succedent to the antecedent via the [/R] rule (Γ_4) in the same way. Finally we get a premise without hypothetic types in the antecedent with a 1-interval antecedent for which the induction hypothesis holds. This case was treated in the previous part as case 1.
If we restrict ourselves to premises that are on three intervals we will still find all proofs. The normal form is complete.

3 Two algorithms

3.1 Backward chaining

A simple algorithm to find out whether or not there is a proof for some theorem is the following.

- If the theorem is an axiom we have a proof.
- If the theorem is not an axiom we try to match it against the conclusion of a right rule. We have one premise now that has to be proved.
- If the theorem is not an axiom and we can not match it against the conclusion of a right rule, we try to match it against the conclusion of a left rule. We have two premises now that have to be proved.

We call this method backward chaining of the logical rules. The rules show that the number of connectives will decrease with every backward application of a logical rule. The connectives are removed one by one. This garantuees that the algorithm will stop. The search space is finite because the number of connectives decreases with every application of a rule.
This is the standard algorithm for Lambek theorem proving. It shows that theorem proving in Lambek calculus is decidable. If we use the subcategorization list notation and allow only premises which are on three intervals we almost have a polynomial algorithm. The only thing we have to add is *memoization* [Cormen, Leiserson and Rivest, 1990, pp. 312-314]. Memoization means that we put in a table the result of our first attempt to compute the solution of a problem. If we later want to solve this problem again we can see the answer immediately in the table.
In this algorithm we store the results of the third step in the algorithm. If we know whether some sequent on which no right rules are applicable is provable or not we store that in the table. When we try to prove the same sequent again we know the answer immediately because it is in the table.
Take for n the number of connectives in a sequent. We have to compute $\mathcal{O}(n^6)$ times whether a sequent with a basic succedent is provable or not. This bound has been shown earlier. In a computation of validity of a (sub) sequent with basic succedent we have to prove at most $\mathcal{O}(n^2)$ pairs of premises. In the worst

case $\mathcal{O}(n)$ right rules have to be applied before the succedent of a premise is basic and the premise can be looked up in the table (or computed). Table look-up can be done in constant time. In time $\mathcal{O}(n^3)$ we can compute (or look up) all subgoals of a goal. The total complexity of the algorithm is $\mathcal{O}(n^6 \times n^3)$ is $\mathcal{O}(n^9)$.

3.2 Forward chaining

In the previous section we saw an algorithm that works from the goal sequent towards the axioms. In this section we are going to do the reverse: we start with axioms and try to derive sequents from them. While we are trying this we continuously keep in mind the goal sequent we are trying to prove. This helps us to reduce the search space. In the previous part of this paper we saw how a unique set of leaves can be assigned to all sub-sequents (premises) in a proof. With the goal sequent in mind it is also possible, however, to find the unique sub-sequent if we know the set of leaves. This is a result of the subformula property. Note that the cardinality of the set of leaves is twice the number of axioms we need to derive the corresponding sub-sequent.
The algorithm is defined as follows. We start with all axioms. There is only a quadratic number of possible axioms. Axioms have a set of leaves of size two. We build sets containing more than two leaves as follows:

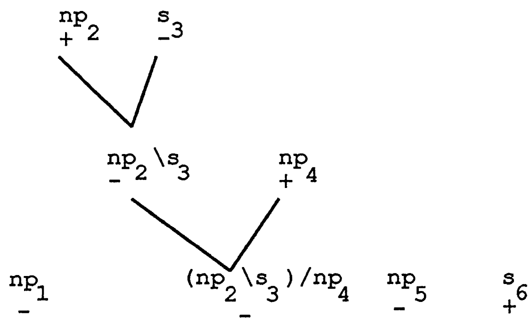
- Take two disjunct sets of leaves that have been found already.
- Test planarity in the unfolded proof frame (cf. [Roorda, 1991] of the goal sequent for the union of the sets.
- Apply a left rule with premises the sequents associated with the two sets of leaves. The type you introduce in the conclusion must be a subtype[2] of the types in the goal sequent.
- Apply right rules on the new sequent of the previous step as often as possible. Again only introduction of subtypes is allowed.

The algorithm builds sets of 4 leaves in the first step, sets of 6 leaves in the next step and so on. We build provable theorems with induction to the number of axioms we need to derive them. Because premises are derived from less axioms than the conclusion we find all theorems.
When we are finished we check whether the goal sequent has been found in the last step.
The number of sequents with a set of $2m$ leaves is polynomial in m as a consequence of the 3-intervals property. If we want to build a sequent of $2n$ leaves, we try to combine sequents of 2 leaves with sequents of $2(n-1)$ leaves, sequents of 4 leaves with sequents of $2(n-2)$ leaves and so on. We try n times to combine a polynomial amount of possible sequents with another polynomial amount of possible sequents. Therefore this algorithm is a polynomial algorithm and it

[2]The notion of subtype must be slightly changed if we use the subcategorization list notation.

finds a proof in three-intervals normal form if there is one.

Suppose e.g. we want to prove: $np, (np\backslash s)/np, np \vdash s$
We unfold the sequent according to the unfolding rules. The leaves of the trees in the proof frame are numbered from left to right in order to make them different[3]:



We start with sets of two leaves.
In the sequel we will show sets of leaves with their unique theorem.

$(\{np_1, np_2\}, np_1 \vdash np_2)$
$(\{np_1, np_4\}, np_1 \vdash np_4)$
$(\{np_2, np_5\}, np_5 \vdash np_2)$
$(\{np_4, np_5\}, np_5 \vdash np_4)$
$(\{s_3, s_6\}, s_3 \vdash s_6)$

sets containing four leaves:

$(\{np_1, np_2, s_3, s_6\}, np_1\ (s_3,[np_2],[\ ]) \vdash s_6)$
$(\{s_3, np_4, np_5, s_6\}, (s_3,[\ ],[np_4])\ np_5 \vdash s_6)$

sets containing six leaves:

$(\{np_1, np_2, s_3, np_4, np_5, s_6\}, np_1\ (s_3,[np_2],[np_4])\ np_5 \vdash s_6)$

In the last step we find the theorem we had to prove so the sequent was valid.
We can not derive this sequent:

$(\{np_2, s_3, np_5, s_6\}, np_5\ (s_3,[np_2],[\ ]) \vdash s_6)$

[3]In order to save space we do not draw the proof frame circular here.

because combination of ($\{\mathrm{np}_2, \mathrm{np}_5\}$, $\mathrm{np}_5 \vdash \mathrm{np}_2$) and ($\{\mathrm{s}_3, \mathrm{s}_6\}$, $\mathrm{s}_3 \vdash \mathrm{s}_6$) would violate the planarity condition.
This algorithm is polynomial but an upper bound is harder to estimate than in the backward chaining algorithm.

4 Discussion

We have given polynomial algorithms for deciding provability in the second order fragment of Lambek calculus. The most important idea is the normal form which restricts possible sub-sequents to those that are on three intervals (or less). Once we have the normal form, the polynomial algorithms are easy to find. The algorithms we gave are both instances of *dynamic programming* [Cormen, Leiserson and Rivest, 1990, Chapter 16].
We can show that the condition of three intervals does not work in higher order fragments (> 2). The type $((c\backslash c)\backslash b)\backslash b$ has order 3. Consider the sequent:

$$c\backslash b, (c\backslash b)\backslash b, ((c\backslash c)\backslash b)\backslash b, ((c\backslash c)\backslash b)\backslash b, ((c\backslash c)\backslash b)\backslash b \vdash b$$

A necessary subproof is:

$$c, c\backslash c, c\backslash c, c\backslash c, c\backslash b \vdash b$$

This sub-sequent is on 4 intervals. By repeating the type $((c\backslash c)\backslash b)\backslash b$ we can construct a counterexample to any fixed bound on the number of intervals one needs in a proof.
In order to find a polynomial algorithm for the whole calculus (so not for the second order fragment only) we have to do something different. It is worth noting that the cut rule is an admissable rule in the Lambek calculus. Algorithms which make use of the cut rule have gained interest since Pentus has proved that the Lambek calculus is context free [Pentus, 1993].
We have given an $\mathcal{O}(n^9)$ algorithm. Maybe it is possible to improve the algorithm such that the bound is decreased. The main goal in this paper however was to show a polynomial time algorithm.

References

Buszkowski, W., On generative capacity of the Lambek calculus, paper presented at JELIA '90: Logics in AI, Amsterdam, 1990.

Cormen, T. H., C. E. Leiserson and R. L. Rivest, *Introduction to algorithms*, MIT Press, Cambridge, MA, 1990.

Hendriks, H., Lambek Semantics, paper presented at 8th Amsterdam Colloquium, Amsterdam, 1992.

Lambek, J., The Mathematics of Sentence Structure, *American Mathematical Monthly*, *65*, 154–169, 1958.

Moortgat, M., Categorial Investigations. Logical and Linguistic Aspects of the Lambek Calculus, Universiteit van Amsterdam, Dissertatie, 1988.

Pentus, M., Lambek Grammars are Context-free, paper presented at Workshop on Grammar, Proof theory and Complexity, Amsterdam, 1993.

Roorda, D., Resource Logics: Proof-theoretical Investigations, Universiteit van Amsterdam, Unpublished dissertation, 1991.

The ILLC Prepublication Series

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

1992

Annual Report 1991

Logic, Semantics and Philosophy of Langauge
LP-92-01 Víctor Sánchez Valencia — Lambek Grammar: an Information-based Categorial Grammar
LP-92-02 Patrick Blackburn — Modal Logic and Attribute Value Structures
LP-92-03 Szabolcs Mikulás — The Completeness of the Lambek Calculus with respect to Relational Semantics
LP-92-04 Paul Dekker — An Update Semantics for Dynamic Predicate Logic
LP-92-05 David I. Beaver — The Kinematics of Presupposition
LP-92-06 Patrick Blackburn, Edith Spaan — A Modal Perspective on the Computational Complexity of Attribute Value Grammar
LP-92-07 Jeroen Groenendijk, Martin Stokhof — A Note on Interrogatives and Adverbs of Quantification
LP-92-08 Maarten de Rijke — A System of Dynamic Modal Logic
LP-92-09 Johan van Benthem — Quantifiers in the world of Types
LP-92-10 Maarten de Rijke — Meeting Some Neighbours (a dynamic modal logic meets theories of change and knowledge representation)
LP-92-11 Johan van Benthem — A note on Dynamic Arrow Logic
LP-92-12 Heinrich Wansing — Sequent Caluli for Normal Modal Propositional Logics
LP-92-13 Dag Westerståhl — Iterated Quantifiers
LP-92-14 Jeroen Groenendijk, Martin Stokhof — Interrogatives and Adverbs of Quantification
Mathematical Logic and Foundations
ML-92-01 A.S. Troelstra — Comparing the theory of Representations and Constructive Mathematics
ML-92-02 Dmitrij P. Skvortsov, Valentin B. Shehtman — Maximal Kripke-type Semantics for Modal and Superintuitionistic Predicate Logics
ML-92-03 Zoran Marković — On the Structure of Kripke Models of Heyting Arithmetic
ML-92-04 Dimiter Vakarelov — A Modal Theory of Arrows, Arrow Logics I
ML-92-05 Domenico Zambella — Shavrukov's Theorem on the Subalgebras of Diagonalizable Algebras for Theories containing $I\Delta_0$ + EXP
ML-92-06 D.M. Gabbay, Valentin B. Shehtman — Undecidability of Modal and Intermediate First-Order Logics with Two Individual Variables
ML-92-07 Harold Schellinx — How to Broaden your Horizon
ML-92-08 Raymond Hoofman — Information Systems as Coalgebras
ML-92-09 A.S. Troelstra — Realizability
ML-92-10 V.Yu. Shavrukov — A Smart Child of Peano's
Compution and Complexity Theory
CT-92-01 Erik de Haas, Peter van Emde Boas — Object Oriented Application Flow Graphs and their Semantics
CT-92-02 Karen L. Kwast, Sieger van Denneheuvel — Weak Equivalence: Theory and Applications
CT-92-03 Krzysztof R. Apt, Kees Doets — A new Definition of SLDNF-resolution
Other Prepublications
X-92-01 Heinrich Wansing — The Logic of Information Structures
X-92-02 Konstantin N. Ignatiev — The Closed Fragment of Dzhaparidze's Polymodal Logic and the Logic of Σ_1 conservativity
X-92-03 Willem Groeneveld — Dynamic Semantics and Circular Propositions, revised version
X-92-04 Johan van Benthem — Modeling the Kinematics of Meaning
X-92-05 Erik de Haas, Peter van Emde Boas — Object Oriented Application Flow Graphs and their Semantics, revised version

1993

Logic, Semantics and Philosophy of Langauge
LP-93-01 Martijn Spaan — Parallel Quantification
LP-93-02 Makoto Kanazawa — Dynamic Generalized Quantifiers and Monotonicity
LP-93-03 Nikolai Pankrat'ev — Completeness of the Lambek Calculus with respect to Relativized Relational Semantics
LP-93-04 Jacques van Leeuwen — Identity, Quarrelling with an Unproblematic Notion
LP-93-05 Jaap van der Does — Sums and Quantifiers
Mathematical Logic and Foundations
ML-93-01 Maciej Kandulski — Commutative Lambek Categorial Grammars
ML-93-02 Johan van Benthem, Natasha Alechina — Modal Quantification over Structured Domains
ML-93-03 Mati Pentus — The Conjoinablity Relation in Lambek Calculus and Linear Logic
ML-93-04 Andreja Prijatelj — Bounded Contraction and Many-Valued Semantics
ML-93-05 Raymond Hoofman, Harold Schellinx — Models of the Untyped λ-calculus in Semi Cartesian Closed Categories
ML-93-06 J. Zashev — Categorial Generalization of Algebraic Recursion Theory
ML-93-07 A.V. Chagrov, L.A. Chagrova — Algorithmic Problems Concerning First-Order Definability of Modal Formulas on the Class of All Finite Frames
ML-93-08 Raymond Hoofman, Ieke Moerdijk — Remarks on the Theory of Semi-Functors
ML-93-09 A.S. Troelstra — Natural Deduction for Intuitionistic Linear Logic
ML-93-10 Vincent Danos, Jean-Baptiste Joinet, Harold Schellinx — The Structure of Exponentials: Uncovering the Dynamics of Linear Logic Proofs
Compution and Complexity Theory
CT-93-01 Marianne Kalsbeek — The Vanilla Meta-Interpreter for Definite Logic Programs and Ambivalent Syntax
CT-93-02 Sophie Fischer — A Note on the Complexity of Local Search Problems
CT-93-03 Johan van Benthem, Jan Bergstra — Logic of Transition Systems
CT-93-04 Karen L. Kwast, Sieger van Denneheuvel — The Meaning of Duplicates in the Relational Database Model
CT-93-05 Erik Aarts — Proving Theorems of the Lambek Calculus of Order 2 in Polynomial Time
Other Prepublications
X-93-01 Paul Dekker — Existential Disclosure, revised version
X-93-02 Maarten de Rijke — What is Modal Logic?
X-93-03 Michiel Leezenberg — Gorani Influence on Central Kurdish: Substratum or Prestige Borrowing