Lambek Calculus and Type Grammars:

Back and Forth

Zhe Lin
ILC Sun Yat-sen University

Joachim Lambek was born in Leipzig on December 5, 1922. His parents moved to Leipzig from a small town near Krakow (Poland). In the late 1930-ties the family left Germany for England. After the outbreak of the SWW, he spend two years in an internment camp in Canada, then entered McGill University in Montreal. He earned Ms.C. in mathematics in 1946. This university has remained his affiliation throughout his academic career; in the years 1963-1993 he was a professor at the Department of Mathematics and Statistics and occupied the Peter Redpath Chair, then a professor emeritus at McGill. He has visited many universities and research centers in Europe and America.

J. Lambek's scientific interests focused on algebra, category theory, mathematical logic, mathematical linguistics and number theory. In category theory he developed categorical logic, e.g. he has shown close connections between cartesian closed categories and typed lambda calculi. In mathematical linguistics, Lambek's Syntactic Calculus, introduced in 1958 (nowadays called Lambek Calculus), is a basic logic for modern type grammars. This calculus and its different variants are extensively studied also in algebraic logic as the basic substrutural logics. In the last decades J. Lambek developed an algebraic approach to grammar, involving pregroups. He also studied some algebras of physics (quaternions) and published books on the history of mathematics.

THE MATHEMATICS OF SENTENCE STRUCTURE*

JOACHIM LAMBEK, McGill University

The definitions [of the parts of speech] *are very far from having attained the degree of exactitude found in Euclidean geometry.*

—Otto Jespersen, 1924.

1. Introduction. The aim of this paper is to obtain an effective rule (or algorithm) for distinguishing sentences from nonsentences, which works not only for the formal languages of interest to the mathematical logician, but also for natural languages such as English, or at least for fragments of such languages. An attempt to formulate such an algorithm is implicit in the work of Ajdukiewicz.† His method, later elaborated by Bar-Hillel [2], depends on a kind of arithmetization of the so-called *parts of speech*, here called *syntactic types*.‡

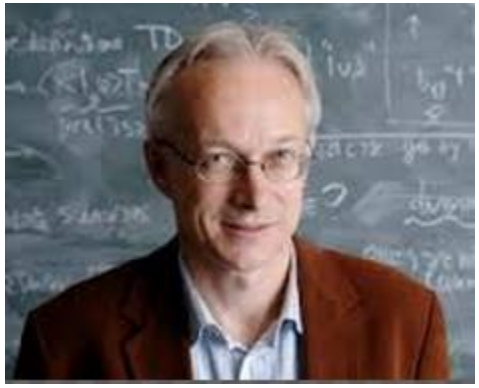
from: Lambek 1958

Poznan Group (W.Buszkowski)

The mathematical research on Lambek systems revived in the early 1980-ties (W.Buszkowski, W. Zielonka, M. Kandulski, and others) investigated logical and computational properties of Lambek systems and categorial grammars, based on them (Lambek grammars); some properties of classical categorial grammars and other type grammars were also studied.

Completeness Results for Lambek Syntactic Calculus (1986)
Generative power of categorial grammars (1986)

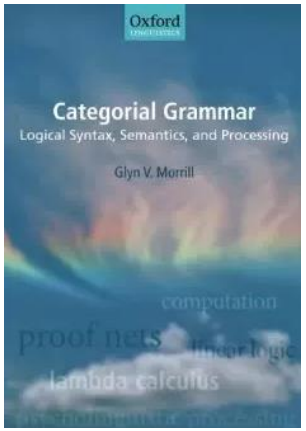
Dutch school (J. van Benthem, F. Zwarts, M. Moortgat)



Dutch school Investigated Lambek systems in relation to the lambda calculus and natural language semantics; there were developments toward general logics of information processing (van Benthem, 1986, 1991, 2005). Moortgat (1996) introduce the modal extension of Lambek calculus and recently the Lambek-Grishin calculus with interesting applications in language description.

van Benthem, Johan. 1986. Essays in Logical Semantics
Multimodal linguistic inference 1996 and Symmetric categorial grammar 2009 M.Moortgat

Barcelona group (Glyn Morrill)



Morrill (1994) elaborated different modal versions of Lambek systems and introduce the discontinuous Lambek calculus basis on applications in language description. He also develop some parsing software for Lambek calculus.

Categorial Grammar: Logical Syntax, Semantics, and Processing

Moscow Group (M.Pentus)

Moscow school Investigated the computational aspects of Lambek Calculus and its variants. M.Pentus solved the long-time open question in this filed (the complexity of the decision problem of Lambek calculus)

Lambek grammars are contextfree (1993)
Lambek calculus is NP-complete (2006)

Other people and results

- Some parallel research was done by scholars adhering to the tradition of Montague Grammar (B. Hall-Partee, E. Bach, R. T. Oehrle) and the Curry-Shaumyan combinatory grammars (M. Steedman, A. Szabolcsi). These works are well documented in collection volumes (Buszkowski et al., 1988; Oehrle et al., 1988).

- Abstract Categorial Grammars(an explicit application of the typed lambda-calculusasa grammar formalism, due to de Groote (2001)), proof nets (a graph-theoretic representation of proofs in multiplicative linear logics), modal categorial grammars, combinatory grammars, and learning theory. For some information on these topics see survey articles (Moortgat, 1997; Buszkowski, 1997, 2003b); also see the books cited above, the collection (Casadio et al., 2005), and two special issues of Studia Logica: 71.3 (2002) and 87.2-3 (2007)

Contents

- An Introduction to the Basic Apspects
- Logics related to Lambek calculus
- Linguistic analysis
- Generative power and complexity
- Other topics
- An open question

Lambek Calculus

An Introduction to the Basic Apspects

Associative Lambek Calculus **L***

$$(\mathrm{Id})\, A \Rightarrow A$$

$$(\cdot\mathrm{L})\ \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, A \cdot B, \Delta \Rightarrow C}, \quad (\cdot\mathrm{R})\ \frac{\Gamma \Rightarrow A;\ \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \cdot B}$$

$$(\rightarrow\mathrm{L})\ \frac{\Gamma, B, \Delta \Rightarrow C;\ \Phi \Rightarrow A}{\Gamma, \Phi, A \rightarrow B, \Delta \Rightarrow C}, (\rightarrow\mathrm{R})\ \frac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B}$$

$$(\leftarrow\mathrm{L})\ \frac{\Gamma, B, \Delta \Rightarrow C;\ \Phi \Rightarrow A}{\Gamma, B \leftarrow A, \Phi, \Delta \Rightarrow C}, \quad (\leftarrow\mathrm{R})\ \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow B \leftarrow A}$$

$$(\mathrm{CUT})\ \frac{\Gamma, A, \Delta \Rightarrow B;\ \Phi \Rightarrow A}{\Gamma, \Phi, \Delta \Rightarrow B}$$

LAMBEK (1958) : **L**, $\Gamma \neq \epsilon$ in $(\rightarrow\mathrm{R})$, $(\leftarrow\mathrm{R})$

(CUT) is admissible in **L**, **L***.

LAMBEK (1961): Nonassociative Lambek Calculus **NL**

Formula structures (trees): formulas, (X, Y); sequents: $X \Rightarrow A$

$$(\cdot\mathrm{L})\ \frac{X[A, B] \Rightarrow C}{X[A \cdot B] \Rightarrow C},\quad (\cdot\mathrm{R})\ \frac{X \Rightarrow A;\ Y \Rightarrow B}{(X, Y) \Rightarrow A \cdot B}$$

$$(\rightarrow\mathrm{L})\ \frac{X[B] \Rightarrow C;\ Y \Rightarrow A}{X[Y, A \rightarrow B] \Rightarrow C},\quad (\rightarrow\mathrm{R})\ \frac{(A, X) \Rightarrow B}{X \Rightarrow A \rightarrow B}$$

$$(\mathrm{CUT})\ \frac{X[A] \Rightarrow B;\ Y \Rightarrow A}{X[Y] \Rightarrow B}$$

NL* (W.B., BULINSKA): the empty structure Λ

$(X, \Lambda) = (\Lambda, X) = X$

(CUT) is admissible in **NL** (LAMBEK 1961) and **NL***.

A residuated semigroup: $\mathcal{M} = (M, \leq, \cdot, \rightarrow, \leftarrow)$ s.t. $(M, \leq)$ is a poset, $(M, \cdot)$ is a semigroup, and $\rightarrow, \leftarrow$ are binary operations on M, satisfying *the residuation law*:

$$\text{(RES) } ab \leq c \text{ iff } b \leq a \rightarrow c \text{ iff } a \leq c \leftarrow b.$$

A residuated monoid: with identity 1, $a \cdot 1 = a = 1 \cdot a$.

A residuated groupoid: $\cdot$ need not be associative.

An assignment f of formulas in $\mathcal{M}$ (a homomorphism from the formula algebra into $\mathcal{M}$). Extended to sequences of formulas: $f(\epsilon) = 1$, $f(\Gamma, A) = f(\Gamma) \cdot f(A)$, and similarly for structures.

$(\mathcal{M}, f) \models \Gamma \Rightarrow A$ iff $f(\Gamma) \leq f(A)$.

L (resp. **L***) is strongly complete w.r.t. residuated semigroups (resp. monoids). **NL** (resp. **NL***) is strongly complete w.r.t. residuated groupoids (resp. with identity).

The rule EXCHANGE

$$\text{(EXC)}\ \frac{\Gamma, A, B, \Delta \Rightarrow C}{\Gamma, B, A, \Delta \Rightarrow C}$$

L with (EXC) is strongly complete w.r.t. commutative residuated semigroups. Analogous facts hold for **L***, **NL**, **NL***. In these systems $A \rightarrow B \Leftrightarrow B \leftarrow A$ is provable.

The $(\rightarrow, \leftarrow)$–fragment of **L*** with (EXC) amounts to BCI. The analogous fragment of **L** is the Lambek–van Benthem calculus (of semantic types) (VAN BENTHEM 1986).

The identity constant 1

$$\text{(1L)}\ \frac{\Gamma, \Delta \Rightarrow A}{\Gamma, 1, \Delta \Rightarrow A},\ \text{(1R)}\ \Rightarrow 1$$

$$\text{(1L)}\ \frac{X[\Lambda] \Rightarrow A}{X[1] \Rightarrow A},\ \text{(1R)}\ \Lambda \Rightarrow A$$

Lattice operations $\wedge$, $\vee$ and constants $\top$, $\bot$; Full Lambek Calculus **FL**; (ONO 1993, JIPSEN 2004). **FL** is strongly complete w.r.t. residuated lattices, i.e. residuated monoids which are lattices.

Modalities (in linguistics MOORTGAT 1995)

$$(\Diamond\mathrm{L})\ \frac{\Gamma, < A >, \Delta \Rightarrow B}{\Gamma, \Diamond A, \Delta \Rightarrow B},\ (\Diamond\mathrm{R})\ \frac{\Gamma \Rightarrow A}{< \Gamma > \Rightarrow \Diamond A}$$

$$(\Box\mathrm{L})\ \frac{\Gamma, A, \Delta \Rightarrow B}{\Gamma, < \Box A >, \Delta \Rightarrow B},\ (\Box\mathrm{R})\ \frac{< \Gamma > \Rightarrow A}{\Gamma \Rightarrow \Box A}$$

Generalized Lambek Calculus (W.B., M. KOLOWSKA-GAWIEJNOWICZ, M. KANDULSKI): f_i $n-$ary connective, $n \geq 1$, $0 \leq i \leq n$, f_i is the $i-$th residual of f_0. $f_0 = f$ (a multi-modal framework, also related to DUNN 1993).

Structures: formulas, $(X_1, \ldots, X_n)_f$

$$(f\mathrm{L})\ \frac{X[(A_1, \ldots, A_n)_f] \Rightarrow A}{X[f(A_1, \ldots, A_n)] \Rightarrow A},\ (f\mathrm{R})\ \frac{X_1 \Rightarrow A_1; \ldots; X_n \Rightarrow A_n}{(X_1, \ldots, X_n)_f \Rightarrow f(A_1, \ldots, A_n)}$$

$$(f_i\mathrm{L})\ \frac{X[A_i] \Rightarrow B;\ (Y_j \Rightarrow A_j)_{j \neq i}}{X[(Y_1, \ldots, f_i(A_1, \ldots, A_n), \ldots, Y_n)_f] \Rightarrow B},\ (f_i\mathrm{R})\ \frac{(A_1, \ldots, X, \ldots, A_n)_f \Rightarrow A_i}{X \Rightarrow f(A_1, \ldots, A_n)}$$

$\mathcal{L}$ a system of logic

$G = (\Sigma, I_G, S)$ s.t. Σ a finite alphabet, I_G a finite relations between symbols from Σ and formulas (types), S a designated type (the principal type).

$I_G(a) = \{A : (a, A) \in I_G\}$

G assigns type A to the string $a_1 \ldots a_n$, $a_i \in \Sigma$, if there exist types $A_i \in I_G(a_i)$, s.t. $A_1, \ldots, A_n \Rightarrow A$ is provable; the set of all such strings x is denoted by $L(G, A)$. $L(G) = L(G, S)$ is *the language* of G.

If antecedents of sequents are trees, then G assigns types to trees. $L_t(G, A)$ consists of all trees T which arise from structures X s.t. $X \Rightarrow A$ is provable by replacing each leave $A \in I_G(a)$ by a. $L_t(G) = L_t(G, S)$ is *the tree language* of G. $L(G)$ is the yield of $L_t(G)$.

One also considers trees determined by proof trees of sequents.

(1) Context-free grammars (CFG's)

(2) Context-sensitive grammars (CSG's)

(3) Classical categorial grammars (CCG's)

Logic: $(\rightarrow, \leftarrow)$–sequents of **L**. (Id), ($\rightarrow$L), ($\leftarrow$L).

Equivalently: (Id),and $A, A \rightarrow B \Rightarrow B$ and $B \leftarrow A, A \Rightarrow B$, (CUT).

The latter is, essentially, the reduction system **AB** of AJDUKIEWICZ (1935) and BAR-HILLEL, GAIFMAN, SHAMIR (1960).

I_G: John: n_1, passes: $(n_1 \rightarrow s) \leftarrow n_1$, every: $n_1 \leftarrow n$, exam: n

John passes every exam.

$n_1, (n_1 \rightarrow s) \leftarrow n_1, n_1 \leftarrow n, n \Rightarrow n_1, (n_1 \rightarrow s) \leftarrow n_1, n_1 \Rightarrow$

$\Rightarrow n_1, n_1 \rightarrow s \Rightarrow s$

Tree: (John (passes (every exam))).

Functor-argument (FA) structure: $(\text{John } (\text{passes } (\text{every exam})_1)_1)_2$.

CCG's (like most categorial grammars) are *lexical*: logic is common for all languages, the language specification is given by the lexical type assignment I_G only.

(4) Lambek categorial grammars (**L**-grammars)

Logic: **L** (often its $(\rightarrow, \leftarrow)$−fragment)

Generating trees: I_G provides $a_i : A_i$

$A_i \Rightarrow B_i$ provable, $B_1, \ldots, B_n \Rightarrow A$ by **AB**. This determines an FA-structure with yield $a_1 \ldots a_n$.

One can generate all possible FA-structures, whence all possible trees, on the generated strings (W.B. 1988).

CURRY–HOWARD CORRESPONDENCE

Natural Deduction

$$(\rightarrow\text{E})\ \frac{\Gamma \Rightarrow A \rightarrow B;\ \Delta \Rightarrow A}{\Gamma, \Delta \Rightarrow B},\ (\rightarrow\text{I})\ \frac{\Gamma, A, \Delta \Rightarrow B}{\Gamma \Rightarrow A \rightarrow B}$$

The proof of $A_1, \ldots, A_n \Rightarrow A$ represented as $x_1 : A_1, \ldots, x_n : A_n \vdash M : A$, where M a (linear) lambda-term with free variables $x_1, \ldots, x_n$.

ND-proofs determine denotations in a type-theoretic semantics of Montague style.

$e \Rightarrow (e \rightarrow t) \rightarrow t$. $\quad x : e \vdash (\lambda y : e \rightarrow t).\,(yx) : (e \rightarrow t) \rightarrow t$.
The lambda-term denotes the (characteristic function of) family of all properties of the individual assigned to x.

VAN BENTHEM 1986: every sequent provable in the $(\rightarrow, \leftarrow)$–fragment of **L** with (EXC) admits only finitely many different readings (different normal ND-proofs).

STANDARD FRAMES

$\mathcal{M} = (M, \cdot)$ a semigroup. $P(M) = \{X : X \subseteq M\}$.

$X \cdot Y = \{xy : x \in X, y \in Y\}$. $X \rightarrow Y = \{z \in M : X \cdot \{z\} \subseteq Y\}$.
$Y \leftarrow X = \{z \in M : \{z\} \cdot X \subseteq Y\}$.

$P(\mathcal{M}) = (P(M), \subseteq, \cdot, \rightarrow, \leftarrow)$ a residuated semigroup. If $\mathcal{M}$ a monoid, then $P(\mathcal{M})$ a residuated monoid with identity $\{1\}$. $P(\Sigma^+)$ the algebra of $\epsilon-$free languages on Σ. $P(\Sigma^*)$ the algebra of languages on Σ.

L (also with $\wedge$) is strongly complete w.r.t. powerset frames over semigroups, and similarly for **L*** and powerset frames over monoids (W.B. 1986). The $(\rightarrow, \leftarrow, \wedge)-$fragments are strongly complete with respect to powerset frames over free semigroups and monoids, respectively.

L is weakly complete w.r.t. algebras of ϵ−free languages, and similarly for **L*** and algebras of languages (PENTUS 1993, 1996).

Proofs are involved; they employ the completeness w.r.t. special relation frames (see below) and some complexity measures of formulas.

NL is not weakly complete w.r.t. powerset frames over free groupoids (tree models) (DOSEN 1994). Soundness and completeness holds for some extensions of **NL** w.r.t. special classes of tree models (VENEMA 1994, 1996).

The $(\rightarrow, \leftarrow)$−fragment of **NL** is (strongly) complete with respect powerset frames over free groupoids (KANDULSKI 1988).

RELATION FRAMES

$P(U^2)$ (a square relation algebra) is a residuated monoid with:

$R \circ S = \{(x, y) \in U^2 : \exists z((x, z) \in R \text{ and } (z, y) \in S)\}$.
$R \to S = \{(x, y) \in U^2 : R \circ \{(x, y)\} \subseteq S\}$.
$S \leftarrow R = \{(x, y) \in U^2 : \{(x, y)\} \circ R \subseteq S\}$. $I_U = \{(x, x) : x \in U\}$.

Relativized frames $P(T)$, where T a transitive relation. In definitions of $R \to S$, $S \leftarrow R$ write $(x, y) \in T$. They are residuated semigroups.

L (also with $\wedge$) is strongly complete w.r.t. frames $P(T)$, where T is an irreflexive, transitive relation, and similarly for **L*** and frames $P(U^2)$. (ANDREKA and MIKULAS 1994).

Related results, e.g. for **NL**, total orderings T, and others (KURTONINA 1995, W.B. and KOLOWSKA-GAWIEJNOWICZ 1997, SZCZERBA 1998, W.B. 2003).

Logics related to Lambek calculus

Linear Logic (Multiplicative intuitionistic linear logic)

Introduced by Jean-Yves Girard in 1987 [Gir87].
Linear logic is:

- Sequent calculus without weakening and contraction.
- As (or more) constructive than intuitionistic logic, while maintaining desirable features of classical logic.
- Finding more and more applications in theoretical computer science.

Language of Linear Logic

- Propositional variables: $A, B, C, \cdots, P, Q, R, \cdots$
- Constants:
  - Multiplicative: $\mathbf{1}, \bot$ (units, resp. of $\otimes, \parr$)
  - Additive: $\top, \mathbf{0}$ (units, resp. of $\&, \oplus$)
- Connectives:
  - Multiplicative: $\otimes, \parr, \multimap$
  - Additive: $\&, \oplus$
- Exponential modalities: $!, ?$
- Linear negation: $(\cdot)^{\bot}$

We now consider the $(\otimes, \multimap, \mathbf{1})$-fragment, **multiplicative intuitionistic linear logic**.

$$\text{(Ax)}\ \frac{}{P \vdash P} \qquad \text{(Ex)}\ \frac{\Gamma, P, Q, \Delta \vdash C}{\Gamma, Q, P, \Delta \vdash C}$$

$$\text{(Cut)}\ \frac{\Gamma \vdash P \qquad P, \Delta \vdash Q}{\Gamma, \Delta \vdash Q}$$

$$\text{(1-R)}\ \frac{}{\vdash \mathbf{1}} \qquad \text{(1-L)}\ \frac{\Gamma \vdash P}{\Gamma, \mathbf{1} \vdash P}$$

$$\text{($\otimes$-R)}\ \frac{\Gamma \vdash P \qquad \Delta \vdash Q}{\Gamma, \Delta \vdash P \otimes Q} \qquad \text{($\otimes$-L)}\ \frac{\Gamma, P, Q \vdash R}{\Gamma, P \otimes Q \vdash R}$$

$$\text{($\multimap$-R)}\ \frac{\Gamma, P \vdash Q}{\Gamma \vdash P \multimap Q} \qquad \text{($\multimap$-L)}\ \frac{\Gamma \vdash P \qquad Q, \Delta \vdash R}{\Gamma, P \multimap Q, \Delta \vdash R}$$

Compact Blinear Logic (pregroup grammars)

Under the influence of Girard (1987), Abrusci (1991) and Lambek (1993) developed a non-commutative version of his linear logic, called

non-commutative linear logic (Abrusci)

or classical bilinear logic (Lambek).

It can be obtained from the Syntactic Calculus by adjoining a constant symbol 0 satisfying:

$$0/(A\backslash 0) \leftrightarrow A \leftrightarrow (0/A)\backslash 0$$

where we can also write

$$A\backslash 0 = A^r, \quad 0/A = A^\ell \ .$$

One can show that

$$(\mathrm{B}^r\,\mathrm{A}^r)^\ell \leftrightarrow (\mathrm{B}^\ell\,\mathrm{A}^\ell)^r \ ,$$

for which it is convenient to write $\mathrm{A} \oplus \mathrm{B}$, or simply A+B. We may think of $\oplus$ as the De Morgan dual of $\otimes$: it corresponds to what Girard calls "par" [1]. Here are some theorems of classical bilinear logic, which had been anticipated by Grishin (1983):

$$\begin{array}{rcl}
1^r \leftrightarrow 0 \leftrightarrow 1^\ell & & \\
\mathrm{A}{+}0 \leftrightarrow \mathrm{A} \leftrightarrow 0{+}\mathrm{A} & & \\
(\mathrm{A}{+}\mathrm{B}){+}\mathrm{C} \leftrightarrow \mathrm{A}{+}(\mathrm{B}{+}\mathrm{C}) & & \\
\mathrm{A}^\ell\mathrm{A} \to 0 & , & \mathrm{A}\mathrm{A}^r \to 0 \\
1 \to \mathrm{A}{+}\mathrm{A}^\ell & , & 1 \to \mathrm{A}^r{+}\mathrm{A} \\
\mathrm{A}/\mathrm{B} \leftrightarrow \mathrm{A}{+}\mathrm{B}^\ell & , & \mathrm{B}\backslash\mathrm{A} \leftrightarrow \mathrm{B}^r{+}\mathrm{A} \\
(\mathrm{A}{+}\mathrm{B})\,\mathrm{C} \to \mathrm{A}+\mathrm{B}\mathrm{C} & , & \mathrm{C}\,(\mathrm{B}{+}\mathrm{A}) \to \mathrm{C}\mathrm{B}+\mathrm{A}
\end{array}$$

The last two are the *mixed associative* laws of Grishin

Lambek observed that there was a simplification if one assumed that

$$\mathrm{A}+\mathrm{B} \leftrightarrow \mathrm{AB} \quad , \quad 0 \leftrightarrow 1 \ .$$

The word "compact" had been used e.g. by Barr (1979) to describe this situation in a categorical context, though then still restricted to the commutative case. Finally, it was realized that *compact* bilinear logic allowed a simpler description as follows:

$$(\mathrm{AB})\mathrm{C} \leftrightarrow \mathrm{A}(\mathrm{BC}) \quad , \quad \mathrm{A}1 \leftrightarrow \mathrm{A} \leftrightarrow 1\mathrm{A}$$
$$\mathrm{A}\,A^r \to 1 \to A^r\mathrm{A} \quad , \quad A^\ell\,\mathrm{A} \to 1 \to \mathrm{A}\,A^\ell \ .$$

Barr, M. (1979). $*$-*Autonomous Categories.* Springer LNM 752, Berlin.

Models in which $\rightarrow$ stands for a partial order are called "pregroups";

they reduce to groups if the order is discrete, that is, is the equality relation, and to partially ordered groups in the so-called *cyclic* case when $A^\ell \leftrightarrow A^r$.

If however the arrows are allowed to stand for morphisms in a category, one would also demand that the above occurrences of $\leftrightarrow$ represent isomorphisms and that the composite arrows

$$A \rightarrow A\,A^\ell A \rightarrow A \quad , \quad A \rightarrow A\,A^r A \rightarrow A$$

are identity arrows, making A^ℓ the left adjoint and A^r the right adjoint of A in a 2-category.

A *pregroup* $\{\mathrm{G}, \cdot\ , 1, {}^{\ell}, {}^{r}, \to\}$ is a partially ordered monoid in which each element a has a

$$\textit{left adjoint } a^{\ell} \quad \text{and} \quad \text{a} \quad \textit{right adjoint } a^{r}$$

such that

$$a^{\ell}\, a \to 1 \to a\, a^{\ell}$$
$$a\, a^{r} \to 1 \to a^{r}\, a$$

the dot "$\cdot$" stands for multiplication with unit 1, and the arrow denotes the partial order.

In linguistic applications the symbol 1 stands for the empty string of types and multiplication is interpreted as concatenation.

Adjoints are unique and we prove

$$1^\ell = 1 = 1^r \,,$$

$$(a \cdot b)^\ell \;=\; b^\ell \cdot a^\ell \quad , \quad (a \cdot b)^r \;=\; b^r \cdot a^r \,,$$

$$\frac{a \to b}{b^\ell \to a^\ell} \quad , \quad \frac{a \to b}{b^r \to a^r} \quad , \quad \frac{b^\ell \to a^\ell}{a^{\ell\ell} \to b^{\ell\ell}} \quad , \quad \frac{b^r \to a^r}{a^{rr} \to b^{rr}} \quad .$$

The following also hold

$$a^{r\,\ell} = a = a^{\ell r} \,,$$
$$a^{\ell\ell}\ a^\ell \to 1 \to a^\ell\ a^{\ell\ell} \,,$$
$$a^r\ a^{rr} \to 1 \to a^{rr}\ a^r \,,$$

A pregroup is *freely generated* by a partially ordered set of *basic* types. From each basic type a we form *simple* types by taking single or repeated adjoints:

$$\ldots\ a^{\ell\ell},\ a^{\ell},\ a,\ a^{r},\ a^{rr}\ldots$$

Compound types or just *types* are strings of simple types. We assign to each word or word form in the dictionary of the language under investigation one (or more) types. The only computations required are *contractions* (C) and *expansions* (E) :

$$(\mathsf{C}) \quad a^{\ell}\ a \to 1\ ,\quad a\ a^{r} \to 1\ ,$$
$$(\mathsf{E}) \quad 1 \to a\ a^{\ell}\ ,\quad 1 \to a^{r}\ a\ ,$$

where a is a simple type. For the purpose of sentence verification expansions are not needed, but only contractions, combined with some rewriting induced by the partial order.

Example	
basic types	$\mathrm{s}\ ,\ i\ ,\ \pi\ ,\ \mathrm{o}\ ,\ \omega,\ \lambda$
compound types	$(\pi_1^r\ \mathrm{s}_1\ \mathrm{o}^{\ell})\ ,\ (i\ \lambda^{\ell})\ ,\ i\ \omega^{\ell}\mathrm{o}^{\ell}$

Developing a pregroup grammar for a natural language consists in two main steps:

(i) assign one or more (basic or compound) types to each word in the dictionary;

(ii) check the grammaticality and sentencehood of a string of words by a calculation on the corresponding types

where the only rules involved are :

contractions,

ordering postulates taking the form $\alpha \rightarrow \beta$ (α, β basic types)

and appropriate conditions introduced in the lexicon, called **metarules**.

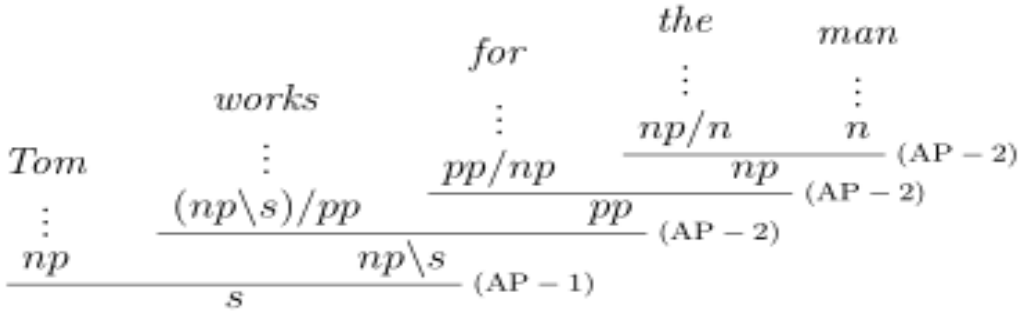
Linguistic analysis

We show some examples of linguistic analysis using AB-grammars. Assume for example primitive types *pp* (prepositional phrase), s (sentence), n (common noun) and np (noun phrase). Consider the following lexicon.

Word	Type(s)
Tom	np
likes	$(np\backslash s)/np$
himself	$((np\backslash s)/np)\backslash(np\backslash s)$
for	pp/np
works	$(np\backslash s)/pp$
man	n

Then '*Tom likes himself*' and '*Tom works for the man*' are derived as sentences as follows.

$$\dfrac{\begin{array}{ccc} Tom & likes & himself \\ \vdots & \vdots & \vdots \\ np & (np\backslash s)/np & ((np\backslash s)/np)\backslash(np\backslash s) \end{array}\ \begin{array}{c} \\ \\ (\mathrm{AP}-1) \end{array}}{\dfrac{\begin{array}{cc} \\ np & np\backslash s \end{array}\ \begin{array}{c} \\ (\mathrm{AP}-1) \end{array}}{s}}$$



Since the sequent $n, (np\backslash s)/pp, pp/np, ((np\backslash s)/np)\backslash(np\backslash s) \Rightarrow s$ is not provable in AB, the AB-grammar (with types as above) does not accept '*Tom works for himself*' as a sentence. In L (Lambek Calculus), using the composition laws, this sequent reduces to $np, (np\backslash s)/np, ((np\backslash s)/np)\backslash(np\backslash s)$, and latter to s. So L-grammars with the same types accept this sentence.

However there are some shortcomings of L as a type logic. L accepts all possible phrase structures on the accepted strings. Further some desirable type are not provable in L. For example '*and*' of type $(s\backslash s)/s$ can not be lift up to $((np\backslash s)\backslash(np\backslash s))/(np\backslash s)$ as in '*Tom sings and dances*'. The required law is not provable in L.

A linguistic example : PG

Example : Let $P = \{\pi_1, \pi_2, \pi_3, s, s_1, s_2, p_1, p_2, o, ...\}$ with

- π_i : subject (I : π_1, you, we, they : π_2, he, she, it : π_3)
- o : direct object
- n : name or noun phrase ($n \leq \pi_3$ and $n \leq o$)
- s_i : present (i=1) or past (i=2) sentence
- s : correct sentence ($s_1 \leq s$ and $s_2 \leq s$)
- p_i : present (i=1) or past (i=2) participle

he loves her

$\pi_3 \quad \pi_3^1 \underline{s_1} o^{-1} \quad o$

a link between a *and* b

means : $ab \leq 1$

John loves Mary

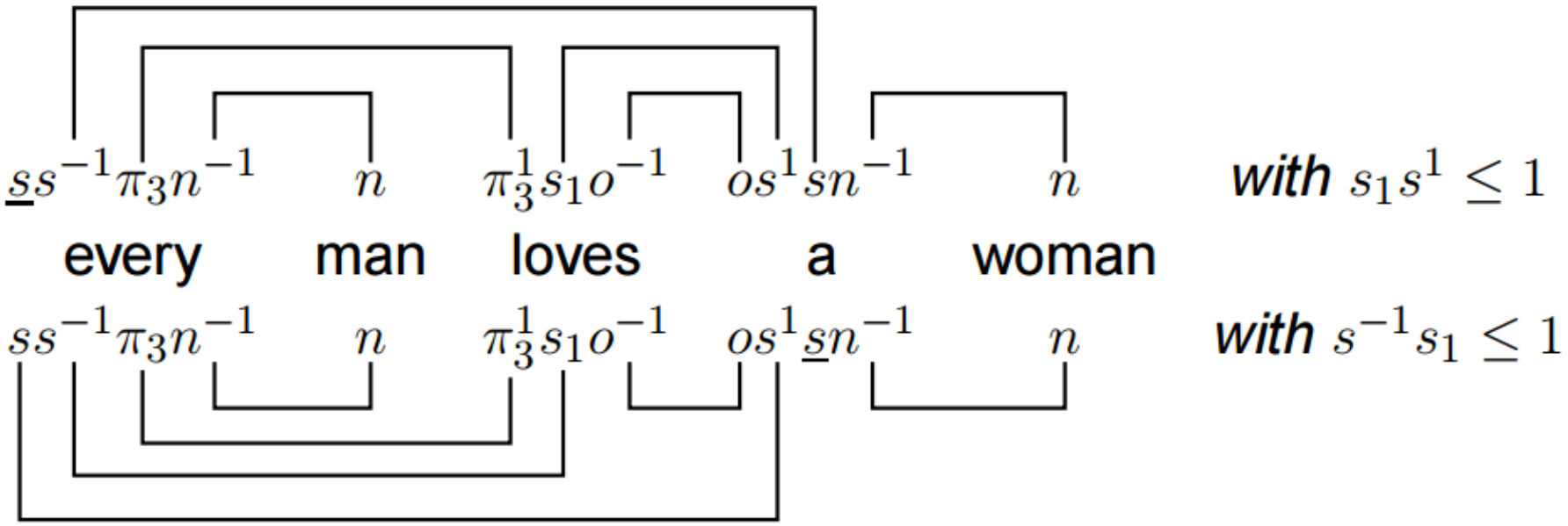
$n \quad \pi_3^1 \underline{s_1} o^{-1} \quad n$

because : $n \leq \pi_3 \Longrightarrow n\pi_3^1 \leq \pi_3\pi_3^1 \leq 1$

and : $n \leq o \Longrightarrow o^{-1}n \leq o^{-1}o \leq 1$

you have been seing her

$\pi_2 \quad \pi_2^1 \underline{s_1} p_2^{-1} \quad p_2 p_1^{-1} \quad p_1 o^{-1} \quad o$


$\underline{s}s^{-1}\pi_3 n^{-1}$ n $\pi_3^1 s_1 o^{-1}$ $os^1 sn^{-1}$ n
with $s_1 s^1 \leq 1$
every man loves a woman
$ss^{-1}\pi_3 n^{-1}$ n $\pi_3^1 s_1 o^{-1}$ $os^1 \underline{s}n^{-1}$ n
with $s^{-1} s_1 \leq 1$

GENERATIVE POWER AND COMPLEXITY

(1) CCGs are weakly equivalent to $\epsilon-$free CFGs (BGS 1960).

Every CCG is equivalent to a CFG. Straightforward: since **AB** always reduces types, then rules $A, A \rightarrow B \Rightarrow B$, $B \leftarrow A, A \Rightarrow B$ can be restricted to subtypes of types appearing in I_G.

Every $\epsilon-$free CFG is equivalent to a CCG; furthermore, the latter employs types of the form p, $p \leftarrow q$, $(p \leftarrow q) \leftarrow r$ only.

This is nontrivial. Actually, it is equivalent to the Greibach Normal Form theorem for CFGs: every $\epsilon-$free CFG is equivalent to a CFG with production rules of the form: $a \mapsto p$, $aq \mapsto p$, $arq \mapsto p$, where $a \in \Sigma$, p, q, r are variables (proved directly by S. Greibach 1967).

The proof in (BGS 1960) is combinatorial. A logical analysis is given in (W.B. 1988, 1996). **L** derives the types assigned by the CCG G' from the rules of the CFG G. $L(G') \subseteq L(G)$ follows from the strong soundness of **L** w.r.t. frames $P(\Sigma^+)$.

(2) **NL**-grammars are weakly equivalent to $\epsilon-$free CFGs. (without product W.B. 1986, with product KANDULSKI 1988).

Every $\epsilon-$free CFG is equivalent to some **NL**-grammar. Now it is an easy consequence of (1) and the fact that, for types restricted as above, a sequent $\Gamma \Rightarrow p$ is provable in **AB** iff it is provable in **NL** (also **L**, **L***, **FL** and so on); use cut elimination.

Every **NL**-grammar is equivalent to a CFG. Now it is nontrivial, since **NL** can expand types, e.g. $A \Rightarrow (B \leftarrow A) \rightarrow B$, $A \Rightarrow B \rightarrow (B \cdot A)$.

NL restricted to simple sequents $A \Rightarrow B$ can be axiomatized as a term rewriting system, based on rewriting rules, e.g. rewrite A on a positive position in C into $(B \leftarrow A) \rightarrow B$, and conversely for negative positions.

KEY LEMMA: $A_1, \ldots, A_n \Rightarrow B$ is provable in **NL** iff there exist $C_1, \ldots, C_n, C$ s.t. $A_i \Rightarrow C_i$ by reducing rules only, $C_1, \ldots, C_n \Rightarrow C$ by **AB**, and $C \Rightarrow B$ by expanding rules only.

This shows that every **NL**-grammar is equivalent to a CCG (which generates the same trees).

(3) **L**-grammars are weakly equivalent to $\epsilon-$free CFGs (PENTUS 1993).

Every **L**-grammar is equivalent to a CFG.

P a finite set of variables. $|A|$ the total number of variables in A.
$T(P,m)$ the set of all types A on P s.t. $|A| \leq m$.

Binary Reduction Lemma: Let $A_1, \ldots, A_n \Rightarrow A_{n+1}$ be provable in **L**, $n \geq 2$, $A_i \in T(P,m)$, for all $i = 1, \ldots, n+1$. Then, there exist $k < n$ and $B \in T(P,m)$ s.t.:
(i) $A_k, A_{k+1} \Rightarrow B$ is provable in **L**,
(ii) $A_1, \ldots, A_{k-1}, B, A_{k+2}, \ldots, A_n \Rightarrow A_{n+1}$ is provable in **L**.

The proof of Binary Reduction Lemma is based on certain proof-theoretic properties of **L**.

$|\Gamma|_p$ the number of occurrences of p in Γ

(I) Interpolation Lemma (ROORDA 1991): Let $\Gamma, \Phi, \Delta \Rightarrow A$ be provable in **L**. Then, there exists a type B s.t.:
(i) $\Phi \Rightarrow B$ is provable in **L**,
(ii) $\Gamma, B, \Delta \Rightarrow A$ is provable in **L**,
(iii) for any variable p, $|B|_p \leq \min(|\Phi|_p, |\Gamma, \Delta, A|_p)$.

A type is *thin*, if each variable occurs at most once in it. A sequent is *thin*, if it is provable in **L**, each type in this sequent is thin, and each variable occurring in the sequent occurs twice in it.

A type is *thin*, if each variable occurs at most once in it. A sequent is *thin*, if it is provable in **L**, each type in this sequent is thin, and each variable occurring in the sequent occurs twice in it.

(II) Every **L**-provable sequent is a substitution instance of an **L**-provable sequent in which each variable occurs twice.

PENTUS proves Binary Reduction Lemma for thin sequents, using free group models. By(I) and (II), it holds for arbitrary sequents.

Consequently, if G is an **L**-grammar on P and m is the maximal A, for A appearing in I_G, then G is equivalent to a CFG whose production rules are all **L**-provable sequents $A \Rightarrow B$ and $A, B \Rightarrow C$, for $A, B, C \in T(P, m)$.

The cardinality of $T(P,m)$ is exponential in the size of P and m. Accordingly, PENTUS's transformation of an **L**-grammar G into an equivalent CFG is exptime in the size of G.

By reducing SAT to the provability problems for **L**, **L***, Cyclic and Noncommutative **MLL**, PENTUS (2006) proves the NP-completeness of these problems.

The universal membership problem for CFGs is polytime. So, if one provided a polytime transformation of any **L**-grammar into an equivalent CFG, then she would prove P=NP.

W.B. (1982) shows that even the $(\rightarrow)-$fragment of **L** with assumptions of the form $p, q \Rightarrow r$ and $p \rightarrow q \Rightarrow r$ is Σ_1^0-complete, and the corresponding grammars generate all $\epsilon-$free r.e. languages. This also holds for **L***.

(4) Again **NL**.

Interpolation Lemma (JÄGER 2004): Let $X[Y] \Rightarrow A$ be provable in **NL**. Then, there exists a type B s.t.:
(i) $Y \Rightarrow B$ is provable in **NL**,
(ii) $X[B] \Rightarrow A$ is provable in **NL**,
(iii) B is a subtype of a type occurring in $X[Y] \Rightarrow A$.

This yields a new proof of the context-freeness of **NL**-grammars.

The provability problem for **NL** is polytime (AARTS 1995 without product, DE GROOTE 2002 with product). So, we get a polytime transformation of any **NL**-grammar into an equivalent CFG (but not into a strongly equivalent CCG).

(W.B. 2005) uses this kind of interpolation to prove that: (i) the consequence relation for **NL** is polytime, (ii) grammars based on finite theories on **NL** are context-free. This holds for **NL** with (EXC), with modalities, and Generalized Lambek Calculus.

L	CFL	NP-complete
L (/, \)	CFL	NP-complete
NL	CFL	P-time
NL (/, \)	CFL	P-time
FL(MALC)	finite intersections of CFL	PSPACE-complete
LG	intersections and permutation closure of CFL	NP-complete

PG	CFL	P-time
DFL,BFL	?	PSPACE-hardness
LL	REL	undecidable
MALL	finite intersections of CFL	PSPACE-complete
MELL	?	?
DFNL,BFNL	CFL	PSPACE-Com

Other topics

(1) Proof nets

A graph-theoretic representation of proofs in multiplicative fragments of substructural logics, introduced by GIRARD (1987) for **MLL**. For noncommutative logics, they are planar graphs. Linguists use them to represent semantic structures of expressions (C. RETORE, P. DE GROOTE, G. PENN, G. MORRILL, R. MOOT). BECHET (2007) gives a new proof of the PENTUS theorem for **L*** (context-freeness), applying proof nets. PENTUS (2006) uses proof nets in his proof of NP-completeness of **L** and related systems.

(2) Bilinear Logic and pregroups

L* is a conservative fragment of Bilinear Logic **BL**; the latter is the multiplicative fragment of both Noncommutative and Cyclic **MLL**. Some authors use **BL** rather than **L*** or **L** as a logic for grammars (V.M. ABRUSCI, C. CASADIO). LAMBEK (1999) introduces a simplified formalism, called Compact Bilinear Logic **CBL**, which arises from **BL** by identifying $\otimes$ and its dual 'par'. Models of **CBL** are called pregroups. This system is essentially stronger than **BL** and incompatible with intuitionistic and classical logics. (W.B., D. BECHET, C. CASADIO, A. PRELLER, N. FRANCEZ, M. KAMINSKI, A. KISLAK-MALINOWSKA)

(3) Unification-based learning

W.B. and PENN (1990) apply the method of unification to design some learning algorithms for categorial grammars (earlier W.B. 1987, VAN BENTHEM 1987). KANAZAWA (1996, 1998) develops and studies these algorithms in direction of Gold's paradigm: learning from positive data. Several authors have obtained interesting results, e.g. D. BECHET, A. FORET, J. MARCINIEC, C. RETORE, C. COSTA FLORENCIO, B. DZIEMIDOWICZ.

(4) Type-theoretic approaches

Many authors prefer to use richer formalisms, e.g. different versions of typed lambda-calculus or higher-order intensional logic. STEEDMAN (1988), following H.B. CURRY, applies certain systems of combinators. RANTA (1994) develops the 'formulas-as-types' paradigm as a type-theoretic grammar. DE GROOTE (2001) introduces Abstract Categorial Grammars, in which both syntactic and semantic structures are represented by lambda-terms, and the two levels are linked by a homomorphism.

An open question

The consequence relations for BCI is decidable?