The Lambek Calculus Enriched with Additional Connectives

MAKOTO KANAZAWA*

Department of Linguistics, Stanford University, Stanford, CA 94305–2150, USA

(Received 11 October, 1991; accepted 28 January, 1992)

Abstract. Some formal properties of enriched systems of Lambek calculus with analogues of conjunction and disjunction are investigated. In particular, it is proved that the class of languages recognizable by the Lambek calculus with added intersective conjunction properly includes the class of finite intersections of context-free languages.

Key words: Lambek calculus, categorial grammar, recognizing power, intersective conjunction, context-free languages

The Lambek calculus (Lambek 1958, van Benthem 1991), 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:

\[
c/b, b/a \Rightarrow c/a, \quad a\backslash b, b\backslash c \Rightarrow a\backslash c.
\]

* 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.

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

\[
a \Rightarrow (b/a)\b, \quad a \Rightarrow b/(a\b),
\]

which allows the functor and the argument to switch roles.

As early as back in 1958, Lambek (1958) 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 \(L(\/, \\) 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 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:**

\[
(\Rightarrow) \quad \frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, b/a, X, Z \Rightarrow c} \quad \quad \left(\Rightarrow /\right) \quad \frac{X, a \Rightarrow b}{X \Rightarrow b/a}
\]

\[
(\Rightarrow) \quad \frac{X \Rightarrow a \quad Y, b, Z \Rightarrow c}{Y, X, a\b, Z \Rightarrow c} \quad \quad \left(\Rightarrow \right) \quad \frac{a, X \Rightarrow b}{X \Rightarrow a\b}
\]

**Cut:**

\[
X \Rightarrow a \quad Y, a, Z \Rightarrow b \quad \quad \frac{X \Rightarrow a \quad Y, X, Z \Rightarrow b}{Y, X, Z \Rightarrow b}
\]

\(X, Y, Z\) stand for sequences of formulas. In \((\Rightarrow /)\) and \((\Rightarrow \)\), \(X\) 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 (Lambek 1958).

The idea of the Lambek calculus is based on a certain natural semantics in terms of sets of expressions (Lambek 1958, 1961, van Benthem 1991). A language model \(L\) assigns 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 \(a\b\) in the following way:

\[
L_{b/a} = L_b/L_a = \{ y \mid \forall x \in L_a \ yx \in L_b \},
\]

\[
L_{a\b} = L_a \backslash L_b = \{ y \mid \forall x \in L_a \ xy \in L_b \}.
\]

\(^1\) In what follows, we assume familiarity with the basic methods and terminology of Gentzen-style proof theory.