Proof-Nets, Hybrid Logics and Minimalist Representations

ALAIN LECOMTE
LORIA-Calligramme, 615 Rue du Jardin Botanique, BP 101, 54602 Villers-les-Nancy, France
E-mail: alain.lecomte@upmf-grenoble.fr

Abstract. In this paper, we aim at giving a logical account of the representationalist view of minimalist grammars by referring to the notion of Proof-Net in Linear Logic. We propose, at the same time, a hybrid logic which mixes one logic (Lambek calculus) for building up elementary proofs and another one for combining the proofs so obtained. Because the first logic is non-commutative and the second one is commutative, this brings us a way to combine commutativity and non-commutativity in the same framework. The dynamic of cut-elimination in proof-nets is used to formalise the move-operation. Otherwise, we advocate a proof-net formalism which allows us to consider formulae as nodes to which it is possible to assign weights which determine the final phonological interpretation.

Key words: generative grammar, hybrid logics, linear logic, proof-nets, type-logical grammar

1. Introduction

The basic idea concerning the use of Proof-Nets (PNs) is to consider words and expressions as building blocks in the construction of proofs of sequents. These building blocks are called modules; they correspond to proof-nets where some premises are mere hypotheses. This conception has many relations to the work on partial proof-trees (PPTs) in the context of Tree Adjoining Grammars (Joshi and Kulick, 1997). Like in the case of PPTs, we are led to hybrid logics in order to give a precise logical formulation of combining PNs: we need a logic for building up elementary proofs and then we need another one for combining these proofs. One of the particularities of our approach is that we shall not use some special rules for combining proofs like stretching in PPTs. Another particularity consists in using proof-nets, whereas in the Joshi-Kulick-Kurtonina approach, it is claimed that Natural Deduction trees exactly provide what is needed for linguistic purposes (Joshi and Kulick, 1997). Our motivation for it is that the proof-net machinery allows a better formalisation of move-operations by means of cut-elimination, where cut-formulae are complex formulae (⊗-conjunctions and ∨-disjunctions). The Minimalist Program (Chomsky, 1996) also makes reference to features which are either weak or strong. This suggests that if we treat features as atomic types (similarly to Cornell, 1996), then these types, when considered nodes in a net, can receive unequal strengths which could explain how variants of the same sentence can be produced.
2. Two Logics

The use of the Lambek calculus (L) with product, in the context of so-called Lambek grammars, requires that the hypotheses be totally ordered. Moreover, its absolute lack of structural rules makes it difficult to reuse a lexical type, even if it is what happens in some linguistic phenomena like cyclic movement. The operation Move, of frequent use in Minimalist Grammars (Stabler, 1997), cannot be conveniently represented in this framework. Let us imagine that, for instance, we want to describe an up- and left-ward movement of a constituent with regards to a verbal head. We would like, then, the d-constituent to be used twice: one time at the position where it is selected by the verbal head, and another time at the position where it receives case. We could think of a product type associated with each determiner phrase, something like: 

\[
\text{d} \times \text{case} \quad \text{(or d \otimes \text{k}, as it will be noted further), but even then the Lambek calculus fails because it cannot express any kind of wrapping. The solution we shall propose to this problem consists of adding an upper level to this rudimentary logic: a level in which it becomes easy to manipulate ready made proofs in L.}
\]

We call a partial proof in a sequent calculus (and later on, a partial proof-net representing this partial proof) a module. A proof is said to be partial if it uses (undischarged) hypotheses.

Let us see, for instance, what could be a “module” associated with a transitive verb, say to like (where d denotes the determiner category, which is a categorial feature, k the requirement for a case-feature, which is a functional feature, and vp the verbal phrase category):

\[
\text{to like :}
\]

\[
\begin{align*}
\text{d} \otimes \text{k} \times (\text{k}(\text{d}\text{vp})/\text{d})^1 \times (\text{k}(\text{d}\text{vp})/\text{d})^1 \times (\text{d} \rightarrow \text{k}(\text{d}\text{vp}))^2 \times (\text{d} \rightarrow \text{k}(\text{d}\text{vp}))^2 \\
(\text{d}\text{vp}) \rightarrow (\text{d}\text{vp})
\end{align*}
\]

This module uses proofs (or, more precisely, conclusions of those proofs) and hypotheses.

\[
\begin{align*}
\text{d} \otimes \text{k} & \text{ is a hypothesis.} \\
(\text{k}(\text{d}\text{vp})/\text{d})^1 & \text{ is “proved” by the lexical item to like.} \\
(\text{k}(\text{d}\text{vp})/\text{d})^1 \cdot \text{d} - \text{o}(\text{k}(\text{d}\text{vp})) & \text{ is a correct deduction in L.} \\
\text{d} \otimes \text{k} \cdot (\text{k}(\text{d}\text{vp}) - \text{d}(\text{d}\text{vp})) & \text{ is the same for (d\text{vp}) - (d\text{vp}).} \\
\text{Indices (1, 2, 3, 4) relate formulae which will be linked by an axiom in the final proof.}
\end{align*}
\]

Finally, this deduction relation says that:

\[
\begin{align*}
\text{If we have the hypothesis d} \otimes \text{k} \\
\text{and proofs of}
\end{align*}
\]

\[
\begin{align*}
\text{((k}(\text{d}\text{vp})/\text{d}), \\
\text{((k}(\text{d}\text{vp})/\text{d}) \cdot \text{d} - \text{o}(\text{k}(\text{d}\text{vp})), \\
\text{k} \otimes \text{(k}(\text{d}\text{vp}) - \text{d}(\text{d}\text{vp}))
\end{align*}
\]

then, by combining them, we can have a proof of (d\text{vp}), with this proof built with axiom links as indicated by the indices. We show this proof in Figure 1 where: