Bi-grammars : a logical system for syntax, semantics and their correspondence

Abstract

After previous proposals concerning a logical account of minimalism, we go deeper into the interface between syntax and semantics. The main idea is that, according to many observations made for instance by Ray Jackendoff, the logical form is not a mere result of a derivation, after formal features consumption, but two tasks are performed on a par : the syntactic analysis on one side and the semantical analysis on the other side. Both analyses are connected by synchronization links in such a way that a parse can crash for (at least) two reasons : either because of a mismatch of syntactic features, or because of a failure in the semantic derivation.

1 Presentation

Ray Jackendoff [2, 3] points out that language is not as *syntactocentric* as generative grammar often says. The new architecture of language proposed as an alternative to syntactocentrism, follows the following guidelines:

> *"[...] there are three independent sources of discrete infinity : phonology, syntax, and semantics. Each of these components is a generative grammar in the formal sense, and the structural description of a sentence arises from establishing a correspondance among structures from each of the three components. Thus the grammar as a whole is to be thought of as a parallel algorithm."*

We here focus on only two of these three components : syntax and semantics, leaving out to a further work a complete picture of such an architecture [1] . We shall therefore assume that both syntax and semantics are global systems, but linked together by means of *synchronization*-links, in such a way that a failure in synchronization results in a derivation crash.

For convenience and elegance, it is preferable to make use of the simplest and best mastered formal systems: we shall therefore take logical ones, which are based on resource-sensitivity. It is no surprise to handle syntax by means of a logical calculus which is resource sensitive to resources and partially to the order among features (for depicting word order), and semantics by means of a a commutative calculus, partially sensitive to the amount of data. The logical system we use for syntax, a slight extension of [1], is quite similar to the famous Lambek calculus [4], which is known to be a neat logical system. This logic under consideration is a super-imposition of the Lambek calculus (a non commutative logic) and of intuitionistic multiplicative logic (also known as Lambek calculus with permutation) and allows for a categorial presentation of minimalist grammars of [6] as shown in our earlier work. The context, that is the set of current hypotheses, is endowed with an order, and this order is crucial for obtaining the expected order on pronounced and interpreted features but it can also be relaxed when necessary: that is when its effects have already been recorded (in the labels) and the corresponding hypotheses can therefore be discharged.

In fact, possible proofs are restricted to those which only use three rules: [$\otimes$ E], [/ E] and [$\backslash$ E] (respectively : tensor elimination, slash elimination and backslash elimination). In earlier versions of our work, we introduced other limitations on proofs : for instance they had to manage the hypotheses according to a particular order (*first in, first out*, and this was needed to express the *shortest move* constraint.[2] In this earlier work semantic representations *à la* Montague were obtained by some transformation performed on the syntactic derivation tree. The *semantic*-proof-trees so obtained were obviously proof-trees in another logical system: ILL (Intuitionistic Linear Logic) and they used two rules, both associated with the linear implication (*introduction* and *elimination*).

[1] For the time being, the phonological part is provided with the syntactic part, as a mere label. Labels are instanciated according to the rule-labelling which, itself, depends on the characteristic weak or strong of the syntactic feature.

[2] In some particular cases, this restriction on syntactic proofs/analyses is still needed (but simpler) in the new model that we propose here; but we hope to replace this limitation by an even closer correspondence between syntax and semantics.

We present here another perspective where both derivations are made on a par. We call it *bi-grammar* because it amounts two have two grammars (very much looking like categorial ones) working in parallel. It is worth to note that

- none of the two systems dominates the other (in accordance with Jackendoff's proposal againts syntactocentrism),
- one system is only concerned by elimination rules : syntax, and the other, semantics, by the two kinds of rules introduction and elimination. That relates to claims made by cognicists [5] according to which introduction rules are rarely used during strongly automatized activities. Assuming they are only used in semantics (which supposes strong connection with the conceptual system) thus makes sense, particularly if their use is tightly constrained by strict syntactic laws expresssed in the "other" system.

2 The grammatical architecture

The general picture of bi-grammars is as follows. A lexicon maps words (or, more generally, items) onto a *pair* of labelled logical formula, where the first component is a labelled syntactic type (a type + a string), and the second one a labelled semantic type (a type + a λ-term). Syntactic types are defined from syntactic or formal features $\mathcal{P}$ (which are propositional variables from the logical viewpoint):

- categorial features (categories) involved in *merge*: $\text{BASE} = \{\mathtt{c}, \mathtt{t}, \mathtt{v}, \mathtt{d}, \mathtt{n}, \ldots\}$
- functional features involved in *move*:
  $\text{FUN} = \{\overline{\mathtt{k}}, \overline{\mathtt{K}}, \overline{\mathtt{wh}}, \ldots\}$

The connectives in the logic for constructing syntactic formulae are the Lambek implications (or slashes) $\backslash, /$ together with the commutative product of linear logic $\otimes$. [3] Semantic types are defined from semantic features **t** and **e**, and they are coindexed with the syntactic type of the same pair.

Example:

$$\begin{array}{lll} reads & ::= & (/reads/ : ((\overline{\mathtt{k}}\backslash \mathtt{t})/\mathtt{vp}))_i \otimes (\epsilon : ((\mathtt{d}\backslash(\overline{\mathtt{k}}\backslash \mathtt{vp}))/\mathtt{d}))_i; \\ & & \lambda N.N(read) : ((\mathbf{e} \to (\mathbf{e} \to \mathbf{t})) \to \mathbf{t}) \to \mathbf{t_i} \end{array}$$

Hypotheses are admitted in both systems (for [$\otimes$ E] in the syntactic one, and for [$\to$ I] in the semantical one), and they correspond one to the other. For instance, an hypothesis corresponding to a categorial feature is associated with an hypothesis in the semantical system (according to some one-to-one mapping, for instance $\mathtt{d}$ corresponds to **e**). Hypotheses associated with functional features are not associated with "semantic" hypotheses: we know that their role is simply to trigger a movement, and thus they have only an indirect effect on the semantics by triggering an abstraction step. In case of head-movement (a case which arises when a lexical entry has as its type a product of functorial types, like it happens mainly with tensed verbs), the first component is associated with a semantic type corresponding to the functorial kind (for instance if $((\mathtt{d}\backslash(\overline{\mathtt{k}}\backslash \mathtt{vp}))/\mathtt{d})$ is the type of the syntactic hypothesis, then $\mathbf{e} \to (\mathbf{e} \to \mathbf{t})$ is the semantic type of the corresponding hypothesis, simply because only $\mathtt{d}$-features indicate true places of arguments, and the second component may correspond to no type in the semantic counterpart, or to some more or less complex type and λ-term if we wish to enrich the semantic representation (for instance by inclusion of properties of aspect, modality and tense).

3 Logico-grammatical rules for merge and phrasal movement

Because of the sub-formula property we need not present all the rules of the system, but only the ones that can be used according to the types that appear in the lexicon. The rules of the syntactic system in Natural Deduction style are the following ones:

[3] The logical system also contains a commutative implication, $\multimap$, and a non commutative product $\bullet$ but they do not appear in the lexicon, and because of the subformula property, they are not needed for the proofs we use.

$$\frac{\Gamma \vdash x : A/B \quad \Delta \vdash y : B}{\Gamma ; \Delta \vdash xy : A}\,[/E]$$

$$\frac{\Delta \vdash y : B \quad \Gamma \vdash x : B\backslash A}{\Delta ; \Gamma \vdash yx : A}\,[\backslash E]$$

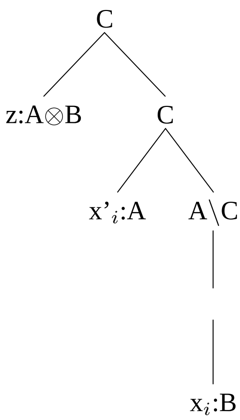
$$\frac{\Gamma[(\Delta_1 ; \Delta_2)] \vdash A}{\Gamma[(\Delta_1 , \Delta_2)] \vdash A}\,entropy$$

$$\frac{\Gamma \vdash \alpha : A \otimes \beta : B \quad \Delta, x : A, y : B, \Delta' \vdash \gamma : C}{\Delta, \Gamma, \Delta' \vdash \gamma[\alpha/x, \beta/y] : C}\,[\otimes E]$$

This later rule encodes movement. It may be noticed that the premisse $\Gamma \vdash \alpha : A \otimes \beta : B$ is not necessarily provided by a lexical entry or by the derivation of some sub-constituant (in which cases, Γ reduces to the empty set). Indeed it can be provided by some instance of the identity axiom, like $z : A \otimes B \vdash z : A \otimes B$. This enables the system to encode **cyclic movement**. (Such new hypotheses correspond on the semantic side to identity types A→A and identity functions).

3.1 Derived Rules and Partial Proof Trees

Actually, the syntactic system will use derived rules that are a combination of several rules. All syntactic derivations can be made by using only these derived rules[4]. We can represent the derived rule in question by a simple partial proof tree :



where moved elements are coindexed.

4 The Semantic System

4.1 Rules

The only rules we have on the semantic side are :

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \to B}\,[\to I]$$

and

$$\frac{\Gamma \vdash u : A \to B \quad \Delta \vdash x : A}{\Gamma, \Delta \vdash u(x) : B}\,[\to E]$$

[4] From the normalisation theorem for this logical calculus, we know that without lost of generality we can assume that [⊗ E] rules only occur immediately after the [/ E] or the [\ E]-step which cancels the feature needed for the application of the [⊗ E] rule.

4.2 The Syntax-Semantic communication

We assume the following things : each step in the partially commutative calculus MG has a counterpart in ILL and reciprocally.

The counterpart of any [$\otimes$ E]-step of the following form:

$$\dfrac{z : A \otimes B \qquad \dfrac{[x' : A] \quad \dfrac{[x : B]}{\gamma : A\backslash C}}{x'\gamma : C}}{z\gamma : C}$$

is a composed step of the following form:

$$\dfrac{z : (T \to U) \to U \quad \dfrac{\dfrac{[x : T]}{\gamma : U}}{\lambda x.\gamma : T \to U}}{z(\lambda x.\gamma) : U}$$

and conversely. The counterpart of any [/ E] or [\ E] step is a [$\to$ E] step, and reciprocally, the counterpart of any [$\to$ E] step is either a [/ E] or a [\ E] step.

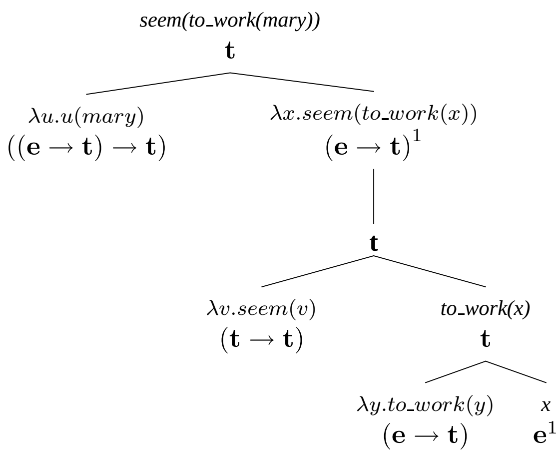
Definition 1 *Two proofs, one in the partially commutative calculus MG and the other in ILL, are said to be synchronized if and only if:*

- *every leaf in ILL has a coindexed counterpart in MG,*
- *steps and their counterparts are performed in the same order in the two proofs*

4.3 Subject raising

Let us look at the example: *mary seems to work* From the lexicon in figure 1 we obtain the deduction tree given in the same figure.

its counterpart is the following semantic tree:



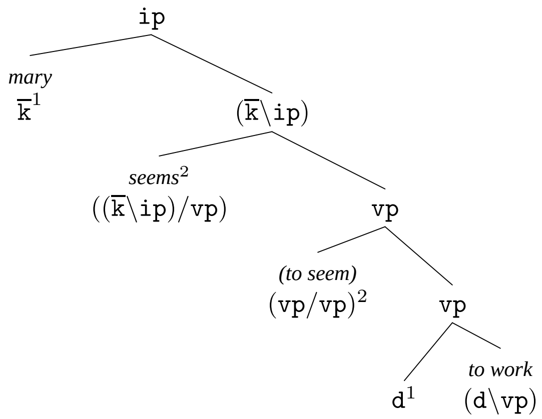
where coindexed nodes are linked by the discharging relation.

5 Conclusion

We present here a parallel architecture for deriving syntactic trees, phonological forms and semantic trees which shows a reciprocal dependency between syntax and semantics where:

Figure 1: Mary seems to work

$$\begin{array}{lll} seems & ::= & \vdash seems : ((\overline{\mathtt{k}}\backslash\mathtt{ip})/\mathtt{vp}) \otimes (\mathtt{vp}/\mathtt{vp}) \\ mary & ::= & \vdash mary : \mathtt{d} \otimes \overline{\mathtt{k}} \\ to\ work & ::= & \vdash to\ work : (\mathtt{d}\backslash\mathtt{vp}) \end{array}$$



- syntax drives semantics : it indicates where introduction rules take place (they are associated with moved constituants),
- semantics drives syntax : it restricts the possible syntactic moves (and thus almost avoids formulating syntactic constraints like shortest move)

The extensive paper will give precise formulations and more examples, in particular sentences which crash for semantic reasons.

References

[1] Philippe de Groote. Partially commutative linear logic: sequent calculus and phase semantics. In Vito Michele Abrusci and Claudia Casadio, editors, *Third Roma Workshop: Proofs and Linguistics Categories – Applications of Logic to the analysis and implementation of Natural Language*, pages 199–208. Bologna:CLUEB, 1996.

[2] Ray Jackendoff. *The Architecture of the Language Faculty*. Number 28 in Linguistic Inquiry Monographs. M.I.T. Press, Cambridge, Massachusetts, 1995.

[3] Ray Jackendoff. *Foundations of Language : Brain, Meaning, Grammar, Evolution*. Oxford University Press, 2002.

[4] Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, pages 154–170, 1958.

[5] Dan Sperber and Deirdre Wilson. *Relevance: Communication and cognition*. Blackwell, 1995.

[6] Edward Stabler. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics, LACL'96*, volume 1328 of *LNCS/LNAI*, pages 68–95. Springer-Verlag, 1997.