Syntax and Semantics interacting in a Minimalist theory

Maxime Amblard[(a)], Alain Lecomte[(b)], Christian Retoré[(a)]
Signes, INRIA-Futurs, Bordeaux
http://www.labri.fr/Recherche/LLA/signes
(a): et Université Bordeaux 1 & LaBRI-C.N.R.S.
(b): et Université Grenoble 2 & CLIPS IMAG-C.N.R.S.

1 Introduction

After several proposals of a logical account of minimalism [5, 4, 6, 1], on the basis of the formalization provided by Edward Stabler [7, 8], we explore more precisely the interface between syntax and semantics. The main idea is that, according to many observations made for instance by Ray Jackendoff [3], the logical form is not the mere result of a derivation after the consumption of formal features. Indeed, there are rather two tasks which are performed on a par : the syntactic analysis properly speaking and the semantical analysis. 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.

2 Rules for syntax

2.1 Rules

We assume the usual elimination rules for / and \, which are the rules of classical categorial grammars. We consider that syntactic features are also linked together by means of a product •, the rules of which are the following ones, where π_1 and π_2 are projections:

$$\frac{\Gamma \vdash x : A \qquad \Delta \vdash y : B}{\Gamma, \Delta \vdash (x, y) : A \bullet B}[i\bullet] \qquad \frac{\Gamma \vdash w : A \bullet B \qquad \Delta, x : A, y : B, \Delta' \vdash z : C}{\Delta, \Gamma, \Delta' \vdash let(x, y) = (\pi_1(w), \pi_2(w))\ in\ z : C}[e\bullet]$$

Remark: if, like it seems natural, • is the product which gives / and \ as its residuates, then it is of course non commutative and in the [e •]-rule, the hypotheses x: A and y:B must be in that order and with no other hypothesis in between them. Let us have a look on the analysis of the VP *see a movie* with the lexicon:

$$\begin{array}{lll} see & : & \vdash /see/ : (acc\backslash \mathtt{v})/\mathtt{d} \\ a & : & \vdash /a/ : (case \bullet \mathtt{d})/\mathtt{n} \\ movie & : & \vdash /movie/ : \mathtt{n} \end{array}$$

Assuming that *acc* is a possible value for *case* and that therefore a type *case* • *t* can discharge hypotheses *acc* (or *nom* or *obl*) and *t*, we obtain:

$$\dfrac{\dfrac{\vdash /a/ : (case \bullet d)/n \qquad \vdash /movie/ : n}{\vdash /a\ movie/ : (case \bullet d)} \qquad \dfrac{y : acc \vdash y : acc \qquad \dfrac{\vdash /see/ : (acc\backslash v)/d \qquad x : d \vdash x : d}{x : d \vdash /see/x : acc\backslash v}}{y : acc, x : d \vdash y/see/x : v}}{\vdash \pi_1(/a\ movie/)/see/\pi_2(/a\ movie/) : v}$$

If a convention (playing the role of the distinction between strong and weak features) intervenes in order to determine the values of π_1, π_2 like for instance the fact that the phonetic value of a *dp* goes to d when *case* is accusative and to *case* when it is nominative, then we shall get:π_1(/a movie/) = ϵ (the empty word) and π_2(/a movie/)= /a movie/, from which follows the deduction of

$$\vdash /see\ a\ movie/) : v$$

This analysis already shows our deductive interpretation of movement: *Move* corresponds to the elimination of the product. Nevertheless, it is clear that if the product is non commutative, then *moves cannot cross each other*. We are thus led to adopt a commutative product with its only residuate, using / and \ as mere variants of the same linear logical implication, only differing on labeling — one can also use the partially commutative calculus of Philippe de Groote [2]. Let us ignore for a while the overgeneration problems such a solution may entail (because from now on, any hypothesis of type h_1 could be discharged at the same time as an hypothesis of type h_2 by a product type $\mathrm{h}_1 \bullet h_2$, and thus in any simple sentence, the object could occupy the position of the subject and vice versa). We set the definition:

Definition 1 *merge = [e \] or [e /] and move = [e •] where / and \ are the same residuate of the commutative product •, simply labeled differently from each other with regards to the way they combine phonetic features.*

2.2 Proof normalization and derived rules

Theoretically, the product elimination steps may occur at any time (as soon as the hypotheses to be discharged have been introduced). The rank of such steps does not matter with regards to the final result. Practically, we shall assume that such steps occur immediately after the needed hypotheses have been introduced. This amounts actually to using a derived rule (with three premises):

$$\frac{\Gamma \vdash w : A \bullet B \qquad x : A \vdash x : A \qquad y : B, \Delta \vdash z : C}{\Gamma, \Delta \vdash let(x, y) = (\pi_1(w), \pi_2(w))\ in\ z : C}\ [e\bullet]^3$$

2.3 Conditions on admissible proofs

Of course all the deductions we can draw in that system are not proper sentence derivations. We previously suggested that with commutativity, the field is open to derivations of non acceptable sentences, or sentences which have not the intended meanings (for instance *Peter loves Mary* as having same meaning as *Mary loves Peter*!). We solve this problem by filtering out the syntactic derivations by the semantic ones. This is done by constructing in parallel a second derivation, based on formal semantic types. The steps of this second derivation must be strictly synchronized with the steps of the first one. For instance, each *merge*-step in the syntactic dimension corresponds to an application step in the semantic one.

3 A type-logical system for semantics

3.1 Semantic types and rules

For the time being, we limit ourselves to semantic types *à la* Montague, i.e. types based on primitive types **e** and **t** by means of only one constructor: $\rightarrow$, which corresponds to the intuitionistic implication.

3.2 Correspondence between syntactic and semantic rules

3.2.1 Move and Cyclic Move

We consider two uses of the rule [e $\bullet$][3]: either the left premise is an extra-logical axiom, or it is an instance of the identity axiom. In the first case a full syntactic object is inserted (either a constituent or a lexical item), where as the second case corresponds to *cyclic moves* (a hypothesis y replaces a previous one x). These two variants correspond to two semantic rules, that we call RAISE and NORAISE, for the reason that the first one asumes a raised type for the moved object while the second one only makes use of the flat (not raised) semantic type.

$$\frac{\Delta \vdash z : T{\rightarrow}U{\rightarrow}V \quad \Gamma \cup [x : T] \vdash \gamma : U}{\Delta \cup \Gamma \vdash z(\lambda x.\gamma) : V}\,[RAISE] \qquad \frac{\Delta \vdash z : T \quad \Gamma \cup [x : T] \vdash \gamma : U}{\Delta \cup \Gamma \vdash (\lambda x.\gamma)(z) : U}\,[NORAISE]$$

3.2.2 Head-Movement

In Chomskyan grammars, head-movement is exemplified by movement from V to I (when verbs raise to their inflection) or from I to C (when auxiliaries raise to the *Comp* position in interrogative sentences). For us, its formulation is:

$$\frac{\Gamma' \vdash \alpha : A\backslash B \bullet \tau(B) \quad \Delta \cup [x' : A\backslash B] \vdash x' : A\backslash B \quad \Gamma \cup [x : \tau(B)] \vdash \gamma : B}{\Gamma', \Delta, \Gamma \vdash \alpha[\epsilon/x]\gamma : A}\,[HM]$$

Because the verbal semantics does not need to be lifted, we shall also use [NORAISE] as its semantic counterpart.

3.2.3 Summary

Figure 3.2.2 sum up the correspondence between semantic and syntactic rules, and bewteen semantic and syntactic types. The translation of syntactic features (like $\overline{\texttt{n}}$, $\overline{\texttt{a}}$, $\overline{\texttt{o}}$) into various semantic types will be clear in the next section. The label *non-var* means that an extra-logical axiom is used to label the syntactic feature, whereas the label *var* means that the identity axiom is used instead. This corresponds respectively to the case where the syntactic feature definitely attracts an item (the movement ends up at this point) and to the case where the item keeps on moving higher (and leftwards). The distinction between $\overline{\texttt{wh}}$ (Q) and $\overline{\texttt{wh}}$ (REL) will be explained by means of examples.

Correspondence Syntax-Semantics	
Syntax	Semantics
ternary-[• E] (non var)	[RAISE]
ternary-[• E] (var)	[NORAISE]
[HM]	[NORAISE]
[/ E] or [\ E]	[→ E]

Correspondence Syntax-Semantics	
Syntax	Semantics
$\mathtt{d}$	$\mathbf{e}$
$\bar{\mathtt{n}}$ (non var)	$(\mathbf{e}_{subj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$
$\bar{\mathtt{a}}$ (non var)	$(\mathbf{e}_{obj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$
$\bar{\mathtt{o}}$ (non var)	$(\mathbf{e}_{indobj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$
$\overline{\mathtt{wh}}$ (Q)(non var)	$(\mathbf{e}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$
$\overline{\mathtt{wh}}$ (REL)(non var)	$(\mathbf{e}\rightarrow\mathbf{t})\rightarrow((\mathbf{e}\rightarrow\mathbf{t})\rightarrow(\mathbf{e}\rightarrow\mathbf{t}))$
$\bar{\mathtt{n}}$ (var)	$\mathbf{e_{subj}}$
$\bar{\mathtt{a}}$ (var)	$\mathbf{e_{obj}}$
$\bar{\mathtt{o}}$ (var)	$\mathbf{e_{indobj}}$

Figure 1: The Syntax-Semantics correspondence

3.2.4 Grammatical functions

We shall assume that objects enter verbal expressions before subjects, thus leading to give transitive verbs the type $\mathbf{e_{obj}}\rightarrow(\mathbf{e_{subj}}\rightarrow\mathbf{t})$ [1]; thus *the first* np *to combine with a transitive verb is necessarily an* object, *and the second one the* subject — this can be generalized to ditransitive verbs.

The correct derivations (syntactic + semantic ones) concerning *Peter kisses Mary* are given in figure 2 (applications of the [• E]-rule are indicated by vertical edges). The lexicon used for this example is :

$$\begin{array}{lll} Peter & \bar{\mathtt{k}}\bullet\mathtt{d} & \lambda P^{(e,t)}.P(Peter) \\ Mary & \bar{\mathtt{k}}\bullet\mathtt{d} & \lambda P^{(e,t)}.P(Mary) \\ kisses & (\mathtt{d}\backslash(\bar{\mathtt{k}}\backslash\mathtt{vp}))/\mathtt{d} & \lambda u^e\lambda v^e K(u,v) \\ (infl) & (\bar{\mathtt{k}}\backslash\mathtt{ip})/\mathtt{vp} & \lambda U^t.U \end{array}$$

The position of $(\mathbf{e}_{obj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$ is due to its correspondence with the accusative case (by figure 3.2.2), and the position of $(\mathbf{e}_{subj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$ is due to its correspondence with the nominative case. To avoid type mismatch, a lambda-abstraction the variable of type $\mathbf{e}_{obj}$ is needed before applying $(\mathbf{e}_{obj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$, and similarly a lambda-abstraction the variable of type $\mathbf{e}_{subj}$ is needed before application of $(\mathbf{e}_{subj}\rightarrow\mathbf{t})\rightarrow\mathbf{t}$.

Thus, the syntactic object only moves to the case position, indexed by ″2″, corresponding to the accusative place, while the syntactic subject moves to the case position indexed by ″1″ corresponding to the nominative place.

3.3 The Syntax-Semantics interface

Let us call $\mathcal{SYN}$ the syntactic calculus with •, / and \, rules [/ E], [\ E], [•E][3] and [HM], and $\mathcal{SEM}$ the semantic calculus with only →, and rules [→E], [RAISE] and [NORAISE]. We assume the following: each step in $\mathcal{SYN}$ has a counterpart in $\mathcal{SEM}$ and reciprocally.
The counterpart of any *ternary*-[• E]-step (Γ' empty) is a [RAISE]-step and reciprocally. The

[1] We have here the choice between considering obj and subj individual constants and $\forall X \mathbf{e}(X)$ a polymorphic type replacing $\mathbf{e}$, or considering $\mathbf{e_{obj}}$ and $\mathbf{e_{subj}}$ subtypes of $\mathbf{e}$.

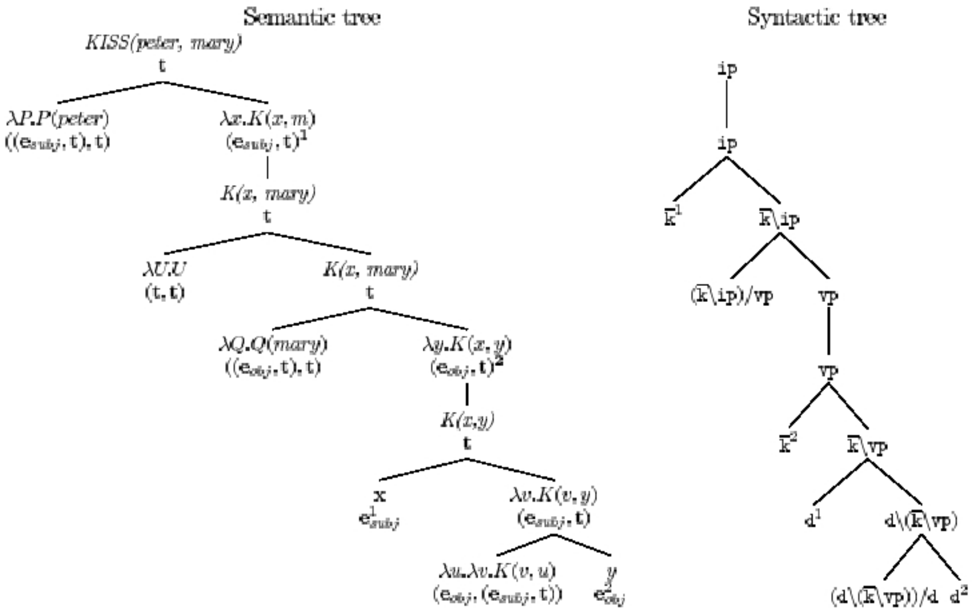

Figure 2: Syntactic and semantic trees for an elementary sentence

counterpart of any *ternary*-[• E]-step with Γ' non empty is a [NORAISE]-step. The counterpart of any [HM]-step is also a [NORAISE]-step. Both [/E] or [\ E] steps correspond to [→E] steps. (cf. fig. 3.2.2).

Definition 2 *Two proofs, one in $\mathcal{SYN}$ and the other in $\mathcal{SEM}$, are said to be synchronized if and only if:*

- *every leaf in $\mathcal{SEM}$ has a coindexed counterpart in $\mathcal{SYN}$,*
- *steps and their counterparts are performed in the same order in the two proofs*

This does not say though how semantical items are distributed among the leaves of the deduction tree in $\mathcal{SEM}$. In fact, semantical items are inserted like phonological features are, during the process of syntactic derivation. The following labeling thus gives the interface between syntax and semantics.

$$\frac{\Gamma \vdash u : A/B \qquad \Delta \vdash x : B}{\Gamma, \Delta \vdash (u\,x) : A}[e/] \qquad \frac{\Delta \vdash x : B \qquad \Gamma \vdash u : B\backslash A}{\Gamma, \Delta \vdash (u\,x) : A}[e\backslash]$$

$$\frac{\vdash (f,u) : A \bullet B \qquad x : A \vdash x : A \qquad y : B, \Delta \vdash z : C}{\Delta \vdash let(x,y) = (f,u)\ in\ z : C}[e\bullet]^3$$

$$\frac{\Gamma \vdash (v,u) : A \bullet B \qquad x : A \vdash x : A \qquad y : B, \Delta \vdash z : C}{\Gamma, \Delta \vdash let(x,y) = (v,u)\ in\ z : C}[e\bullet]^3$$

$$\frac{\Gamma' \vdash (F,G) : A\backslash B \bullet \tau(B) \quad \Delta \cup [x' : A\backslash B] \vdash x' : A\backslash B \quad \Gamma \cup [x : \tau(B)] \vdash \gamma : B}{\Gamma', \Delta, \Gamma \vdash let\ (x',x) = (F,G)\ in\ \gamma : A}[HM]$$

Evaluation by beta-reduction of the formulae obtained by inserting the lambda terms form the lexicon yields the expected result.

4 Conclusion

The precise definition of the correspondence would require to handle lambda-terms with context rather than plain lambda terms. In this precise setting, movement corresponds to lambda-abstraction of the variable which is substituted with the moved term, while movement performs type raising on the term which is substituted. [1]

A comparison can be made with the standard correspondence bewteen categorial grammars (say Lambek grammars) and Montague semantics, since this is basically an extension of this correspondence to a richer syntactic formalism. For instance, what happen when we deal with quantifiers in this minimalist setting? Firstly there is no need to introduce different syntactic types for different syntactic position of the quantifier: movement enables a single syntactic type to apply in various syntactic position, with a perfect correspondence with the semantics of the quantifier. Secondly, there are readings which are possible in the categorial settings but not in this setting — e.g. when the leftmost quantifier is in the scope of a quantifier on its right. It is not easy to tell whether this property of our model is welcome, since some claim that this reading is a topicalisation while others equally accept both readings.

References

[1] Maxime Amblard. Reprsentations smantiques pour les grammaires minimalistes. Mémoire de D.E.A, Université Bordeaux 1, 2003. `http://www.labri.fr/Recherche/LLA/signes`

[2] 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.

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

[4] Alain Lecomte. Rebuilding the minimalist program on a logical ground. In C. Retoré and E. Stabler, editors, *Special Issue on Resource Logics and Minimalist Grammars*, volume 2(1) of *Journal of Research on Language and Computation*, pages 27–55. Kluwer, 2004.

[5] Alain Lecomte and Christian Retoré. Extending Lambek grammars: a logical account of minimalist grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, ACL 2001*, pages 354–361, Toulouse, July 2001. ACL.

[6] Christian Retoré. Semantic aspects of minimalist grammars. In F. Spoto, G. Scollo, and A. Nijholt, editors, *Algebraic Methods in Language Processing – AMiLP 2003*, volume 21 of *TWLT*. University of Twente, 2003.

[7] 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.

[8] Edward Stabler. Remnant movement and structural complexity. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff, and Richard Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Sema ntics*, pages 299–326. CSLI, 1999. distributed by Cambridge University Press.