Syntax-Semantics Interface

One of the ultimate goals of the Sémagramme *project is to provide theories and tools that allow logical representations of the meaning of natural language utterances and discourses to be computed. The construction of such logical representations is syntactically guided, and the question of how these logical representations can be computed from syntactic structures is a crucial one.*

To answer this question, we need to model the syntax-semantics interface, that is, the very process by which the meaning of an utterance may be derived from its syntactic analysis. More precisely, to provide a model of the syntax-semantics interface amounts to answer the following questions:

- *What is the input of the semantic computation? That is, on which kind of (syntactic) structure does the semantic computation operate?*
- *What are the (abstract) mathematical means and principles that are used to compute the semantic form from the syntactic structure?*
- *In which sense is the computation compositional? That is, how is it that the "meaning of the whole" is computed from the "meaning of the parts"?*

In some grammatical theories, the syntax-semantics interface is left implicit, the semantic representation of an utterance being computed at the very time of its syntactic analysis. Typically, in unification grammars, the semantic form is encoded as one of the components of the feature structure. This semantic feature is then built, by unification, at the same time as the other (morpho-syntactic) features.

Nevertheless, defining explicitly a model of the syntax-semantics interface presents several advantages. At the operational level, it allows the semantic analysis to be as independent as possible from the syntactic formalism that is used. Indeed, what matters are the syntactic structures themselves (typically, syntactic trees or dependency graphs), not the way they are obtained from the surface forms. At a more theoretical level, it participates to the study of the compositionality principle. This principle, as a general (rather vague) statement, is difficult to challenge. Nevertheless, the literature on compositionality is quite huge, and provides a lot of examples of semantic phenomena that may be considered not to be compositional. To say that such or such phenomenon is not compositional, however, does not make sense per se. *It only makes sense with respect to a given (sometimes, left implicit) compositionality model. We therefore need explicit compositionality models, and we claim that a compositionality model, from a computational standpoint, is nothing but a model of the syntax-semantics interface.*

0.1 Functional Model

Modern formal semantics, in linguistics, started with the work of Richard Montague at the end of the seventies. Challenging the opinion of a number of philosophers of his time, Montague showed how standard tools from mathematical logic (proof-theory and model-theory) could be used to assign semantics to natural language utterances. His work, which is still a reference today, has given rise to the research domain that is now known as Montague semantics.

In Montague's theory (as it is the case in mathematical logic), semantics is obtained as a homomorphic image of syntax. This corresponds to rather strict view of compositionality, which is known as Montague's homomorphism requirement.

A naive interpretation of this requirement would lead to express both syntax and semantics in first-order structures. Montague's setting, however, is much more powerful. He expresses his logical representations of meaning in a variant of higher-order logic, and uses simply-typed λ-terms to specify the semantic composition operators.

Montague semantics suits quite well the categorial grammar approach whose foundations are also type-theoretic. This has been pointed out at the end of the eighties by van Benthem who showed how the two formalisms could be connected through a syntax-semantics interface based on the Curry-Howard isomorphism. An abstract view of this architecture has lead us to define a new grammatical formalism: *the Abstract Categorial Grammars* (ACG, for short).

An ACG consists of two free higher-order algebras (i.e., two higher-order signatures) related by a homomorphism, which is defined by means of simply-typed linear λ-terms. We claim that the ACG formalism provides an interesting model of compositionality. Nevertheless, this model needs further investigations and developments. In the current model, the terms of the higher-order algebras correspond to simply-typed linear λ-terms. Extending this language by considering other type-theoretic constructions would increase the expressive power of the framework, and allow for finer semantic phenomena to be tackled. Another question concerns the reversibility of the model. Indeed ACGs are such that both syntax and semantics are specified by means of higher-order signatures. It is therefore possible, in principle, to perform both syntactic parsing and semantic analysis in a uniform framework. This raises the question of inverting the homomorphisms (which results in the design of parsing algorithms). Finally, at a more theoretical level, several problems remain open, among others, the decidability of the membership problem, which can be shown to be equivalent to the decidability of the multiplicative exponential fragment of linear logic.

0.2 Relational Model

A consequence of the functional model is that to every syntactic structure corresponds exactly one possible semantic interpretation. This is why in a Montague grammar there cannot be any semantic ambiguity that would not derive from a syntactic ambiguity. Such a point of view is rather drastic, and one would like to relax the model in order to allow for n-to-m-correspondences between syntax and semantics.

There is an easy way out of this problem because every relation may be specified by composing a function with the inverse of another function. In our Montagovian setting this solution would require some intermediate structures together with two homomorphisms: one from the intermediate structures to the syntactic structures, and the other one from the intermediate structures to the semantic forms. This way of proceeding, however, might appear as a pure artefact, and raises several questions. What is the status of these intermediate structures? Do they make sense from a linguistic point of view? If yes, they would appear as a kind of "enriched" syntactic structures, in which case we are just back to the functional model. Moreover, the compositionality paradigm remains algebraic, and is not quite different from Montague's homomorphism requirement.

In fact, Montague's algebraic setting suits well generative grammars whose parse structures are trees (or possibly λ-terms). Other sorts of grammars (typically, dependency grammars and, among them, the Interaction Grammars that have been developed by the Calligramme project) may produce syntactic structures that are plain graphs. A dependency graph is indeed a structure in which syntactic dependencies are closely related to predicate-argument relations. They can be easily enriched with relations derived from the syntax, which are usually ignored (such as arguments of infinitives or syntactically-determined anaphora). As a consequence, the syntactic structure associated to a sentence often appears to be a mere graph rather than a tree. On the semantic side, graphs are also widely used by different formalisms and theories, such as *dependency minimal recursive semantics* or *meaning-text theory*.

We are therefore seeking for a model of the syntax-semantics interface whose both source and target structures are plain graphs. An obvious choice is to appeal to graph rewriting as a computational paradigm. A graph rewriting system is defined as a set of *graph rewrite rules* and a computation is a sequence of rewrite rule applications to a given graph. The application of a rule is triggered via a mechanism of pattern matching, hence a sub-graph is isolated from its context and the result is a local modification of the input. This allows semantic phenomena to be locally tackled, and corresponds to a context sensitive compositionality principle.

Nevertheless, graph rewriting, as a general computational paradigm, is too powerful, and we need to find a good trade-of between expressiveness and efficiency. A first problem to solve consists in identifying the class of graph patterns one wants to rewrite, and to develop efficient graph matching algorithms for this class (in general, in order to apply a graph rewriting step, one needs to solve a sub-graph isomorphism problem, which is too costly). We also intend to explore the possibility of gathering rewriting rules into modules. This would allow the several rewriting steps to be organized into linguistically-motivated consistent sets of rules. Finally, since our goal is to develop a relational model, we intend to deal with rewriting systems that are not confluent. This might be critical and result inefficiencies. Consequently, we want to study local properties of confluence. This should allow to isolate modules corresponding to confluent subsets of rules, non-confluence being confined at the level of module interactions.

0.3 Case studies

As we said, one of our goals in developing syntax-semantics interface models is to obtain formal compositional frameworks. These frameworks will be used as guidelines when modeling dynamic phenomena related to discourse interpretation. It is therefore important to test them against case studies.

Semantics ambiguity is s first case whose modelling crucially relies on the syntax-semantics interface. From an abstract point of view, it describes phenomena where a single syntactic analysis corresond to many semantic representations. Such phenomena occur in particular when the semantic scope of some expressions (such as quantified noun phrases, wh-expressions ot comparative phrases, in the so-called *overt* and *covert* movements) is underdetermined by their syntactic position. The syntax-semantics interface needs to take into account both a relative freeness of the items at a semantic level and a relative strictness of the items at a syntactic level.

A second interesting case relates to idiomatic expressions involving several words but whose meaning might differ drastically from the meaning of its parts (such as *to kick the bucket*, *a red cent*, *casser sa pipe*). Moreover, these locutions differ in their (syntactic) rigidity: some accept passivation, some accept different agents... So a syntax-semantics interface have take into account these variabilities and their effects on the semantic part (the effect of the adjective *bon* in *a lui en a bouch un bon coin* is not to qualify the noun *coin* but rather intensify the event).

When dealing with pronouns and reference, some anaphoric phenomena clearly relate to the dynamicity effects of the discourse. However, some other effects are much closer to the usual syntax-semantics interface. This is in particular the case in sentences with an *obligatory coreference* between a pronoun and a NP (as in *Zelda bores herself*) or an *obligatory non coreference* (as in *she adores Zelda's teacher*). A lot of theses constraints have been studied in Binding Theory, and it would be very interesting to import them into a general model of syntax-semantics interface.