Towards a Type-Theoretical Account of Lexical Semantics

Christian Bassac, Bruno Mery, Christian Retoré

Noname manuscript No.
(will be inserted by the editor)

Towards a Type-Theoretical Account of Lexical Semantics

Christian Bassac · **Bruno Mery** · **Christian Retoré**

the date of receipt and acceptance should be inserted later

Keywords Montague Semantics, Lexical Semantics, Type Theory.

Abstract After a quick overview of the field of study known as "Lexical Semantics", where we advocate the need of accessing additional information besides syntax and Montague-style semantics at the lexical level in order to complete the full analysis of an utterance, we summarize the current formulations of a well-known theory of that field. We then propose and justify our own model of the Generative Lexicon Theory, based upon a variation of classical compositional semantics, and outline its formalization. Additionally, we discuss the theoretical place of informational, knowledge-related data supposed to exist within the lexicon as well as within discourse and other linguistic constructs.

The formalization of the structure of natural language utterances around a surface form (phenogrammatics), a deep structure (tectogrammatics) and the meaning thereof as a logical form (semantics) has developed from the original theories of Curry and Montague to form coherent, type-driven models. Most of these new theories rely upon variations of the compositional analysis of the sentence: from pheno to tectogrammatics, and then to semantics.

Our contribution to this work aims at giving such a model a means to overcome the problems posed by polysemous lexical units during the semantical analysis of the tectogrammatical form.

Building upon an assumed "deep structure", we formalize parts of Pustejovsky's Generative Lexicon Theory, linguistically motivated in [Pustejovsky, 1995], in a pre-processing of the semantics of the sentence. The mechanisms of Lexical Semantics we propose are an additional layer of classical Montague compositional semantics, and, as such, integrate smoothly within such an analysis; we proceed by converting the lexical data to modifiers of the logical form.

This treatment of Lexical Semantics furthermore induces us to think that some sort of non-evident background knowledge of the common use of words is necessary to perform a correct semantic analysis of an utterance. These "commonsense metaphysics" would therefore not be strictly confined to pragmatics, as is often assumed.

INRIA Bordeaux Sud-Ouest, LaBRI (C.N.R.S. & université de Bordeaux)
CRTT (EA 4162 & université de Lyon 2)
LaBRI, 351 cours de la Libération, F-33405 Talence Cedex, France

1 The need for Lexical Semantics

After a brief review of common assumptions occurring in many modern theories of semantics and grammar, we will take polysemy and the creative use of words as a basis to advocate the need for lexical semantics, and place it within the analysis.

1.1 The twofold Montagovian analysis

We assume the existence of a neo-Curryan framework composed of tectogrammatics as the deep structure that yields, on the one hand, the actual written or uttered language, i.e., phenogrammatics, and on the other hand, its intended logical meaning, i.e., semantics. The process of the classical Montagovian analysis of a sentence would be to use the rules of the phenogrammar to infer the tectogrammatical structure from the sentence, by phonological and syntactic means, and then to compute the semantics of the tectogrammatical structure by a type-driven composition of lexemes. Refinements, such as intensionality or modality, might take place, but most systems use a lexicon to associate a semantic type to each word, and let the mechanism operate.

1.2 Polysemy and meaning

The association of single types (and therefore logical behaviour) to lexemes is hampered by the fact that polysemous words in natural language are the rule, not the exception. *Contrastive ambiguity* (the fact that the same sequence of phonemes have come to represent different and unrelated senses during the evolution of the language) led to the construction of lexicons with several possible types for every lexeme, and *ad hoc* mechanisms to select the correct typing in context. But those heuristics are problematic, and, as [Pustejovsky, 1995] points out, *sense-enumeration* lexicons do not solve the problems of *logical polysemy*, in which lexemes might assume many different, yet related senses in context. The need for the speaker to use preliminary, background knowledge of at least some information about every word used appears there clearly: that data enable one to distinguish between the correct and incorrect uses of the word, and to create and recognize new ones. This is not a new idea : it has been expressed in [Searle, 1979], for instance, and many ontology-based approaches of Natural Language Processing assume such a stance.

1.3 Lexical semantics: an additional layer of analysis

More specifically, when computing the semantics for a phrase like *a loving smile*, the speaker (and thus, any processing system) needs to be able to extract some non-trivial information from the words used: here, that a *smile* is typically associated to a person. In the phrase *refreshing tea*, not only do we need to know the properties associated to the plant, *tea*, but also that it might be processed to produce a brewage that can be qualified as *refreshing*. And in order to understand sentences such as *København is a both a seaport and a cosmopolitan capital*, the speaker needs to know that nouns denoting towns and cities might be employed to denote either the associated *geographical* or *demographic* data.

Thus, lexical semantics information is certainly needed in order to process certain utterances correctly. More importantly, the information used here is intrinsically associated with the *word* used, not the context; it is not a matter of pragmatic interpretation. Ideally, therefore, it should be integrated before or during the composition of the logical form.

2 The Generative Lexicon and Montague semantics

After a brief review of some of the points of Pustejovsky's Generative Lexicon Theory, specifically the descriptive, ontology-grounded hierarchy of lexical types, we examine the past and current attempts to integrate portions of this theory in a compositional framework, their strength and shortcomings. We then introduce the principles that have driven our new formalization.

2.1 The Generative Lexicon Theory

The Generative Lexicon is a set of principles that allows lexical composition (the use of words in conjunction with one another) to generate a potentially unbounded number of senses for each lexical unit, from a rich but finite description of the lexemes. Various kinds of logical polysemy, such as alternations, are treated thoroughly. The Generative Lexicon can also account for the *creative use* of words, through the same mechanisms.

2.1.1 Ontological types

Those mechanisms heavily rely upon lexical types. The types associated to lexemes are based upon a hierarchy of concepts, organized in an ontology. Thus, a relation of inheritance is supposed: objects of type *entity* include objects of type *physical* and *abstract*, and everything is included in objects of Montagovian type e.

The construction of such an ontology is a task in itself; Pustejovsky merely supposes its existence.

2.1.2 Rich data structures

To each lexical type is associated a number of structures, which convey the necessary information in order for lexical composition to have the intended effects. On top of the *inheritance structure* needed to build the type hierarchy, there are: the *argument structure*, which conveys the number and types of arguments needed by a predicate, and additional data covering default and optional arguments; the *event structure*, which adds Davidson-style reification to the formalism; and the particularity of the theory, the *qualia structure*.

The *qualia* (singular: *quale*) can be thought of as meta-data, information associated intrinsically to the lexeme, identified as necessary, background knowledge for semantics. Pustejovsky, in accordance to Aristotle, identifies four *qualia*: the `formal`, a set of associated properties (such as color or weight); the `constitutive`, a set of lexemes that the item is typically in meronymic association with (commonly: `made-of` and `part-of` relations); the `agentive`, indicating terms that might cause the lexeme to appear or come into being; and the `telic`, indicating lexemes associated with the purpose, or typical behaviour of the lexeme.

In Pustejovsky's original theory, *qualia* merely indicate the *lexical type* of the items that might fill these roles.

2.1.3 Type coercion, type accommodation

Having detailed the data structures conveying the necessary information, the Generative Lexicon primarily uses operations over types to make use of that information. Semantical composition takes place as usual, with application occuring whenever an argument and a predicate are of compatible types. On top of that, the Generative Lexicon introduces additional mechanisms, used in case of a *type clash* :

- *Type accommodation* is used when the expected type of the argument is an ancestor (with respect to the type hierarchy) of its actual type. The application is then considered valid. (For instance, the predicate *break* applies to all `physical` objects, and thus also to `artifacts`, `vehicles`, and `cars`.)
- *Type coercion* is used when the expected and actual types of the argument are neither identical, nor compatible in the hierarchy. Then, the lexical information is used in order to find a link to the expected type. A common case of type coercion is *qualia exploitation*: if the expected type is that of some quale, then that quale is used instead of the whole argument. Thus, supposing that the predicate *to yearn for* expects an argument of type *event*, and that the telic (purpose) of the object *cigarette* is the event *to smoke [the cigarette]*, then the inference:

  to yearn for a cigarette
  $\rightarrow$ *to yearn for the smoking of a cigarette*

  is made by means of lexical composition.

2.1.4 The •-type construction

Additionally, the Generative Lexicon allows for some specific lexemes to bear more than one type, corresponding to alternate aspects that the term implies: for instance, a *book* is both a physical object (with weight and shape), and an informational content (with an author), and would thus have $P \bullet I$ for type. This is called the •-type (dot-type, or complex type) construction, and words of such a lexical types are called *dot objects*. From the point of view of the formalism, given types A and B, the complex type incorporating both is defined as $A \bullet B$, and might be interpreted either as A, B, or $A \bullet B$.

This particular construction has induced many difficulties and discussions. Its original formulation, *type pumping*, a product with straightforward projections, has been disputed: in the same manuscript, Pustejovsky hints that the •-product is not associative, as the type `journal` would be `society•(physical•information)`: factual evidence points out that meanings representative of both the publisher and physical aspects of a journal are not felicitous. Moreover, later studies have expressed doubts about the projective nature of the operation (a `book` consists of `physical` and `information` aspects, but the pairing of two such arbitrary items does not always result in an actual `book`), and even doubt whether the •-type construction is an actual product.

We therefore use the •-type construction more as the characterization of a category of phenomena to account for than as a well-established operative construction.

2.1.5 Additional lexical operations

Through the study of some specific constructions, several more lexical operations have been added to the Generative Lexicon. *Grinding*, for instance, is a destructive operation turning the reference of some words from a material (such as a plant) into a product (such as food) – e.g., *fruit*. Those operations cannot be handled directly by the general mechanisms expressed above, but their addition fits the formalism neatly.

2.2 Current formulations

Since Pustejovsky made his proposal, many refinements have been proposed. On the one hand, the linguistic theory has been corrected and amended; on the other hand, some formalizations and implementations have been proposed. We review some of the current formulations of the Generative Lexicon Theory, their formal assumptions and shortcomings.

2.2.1 Type operations : the original formulation

[Pustejovsky, 1995], as we have seen, assumes that the composition may directly take place with additional operations (accommodation, coercion, grinding...) acting on type clashes. This is not entirely sufficient for whole sentences, as quantification, anaphora resolution, and variable binding, for instance, are not detailed; it also does not interface neatly with existing compositional formalisms. Moreover, the operation known as *type pumping*, which Pustejovsky uses to model •-types, is logically inadequate, as information that is needed in the further processing of a sentence is lost when it is applied. While formally unsatisfying, that original framework is the basis of an actual implementation in the Object-Oriented paradigm, detailed in [Gupta and Aha, 2003] and [Gupta and Aha, 2005].

2.2.2 Transfers of Meaning

[Nunberg, 1993] proposes a different point of view: the consideration of *meaning transfers*, or shifts. In the author's perspective, the argument does not change its perceived nature when applied to a predicate that does not select for its type. Instead, the *predicate* changes its meaning, in order to accomodate for the argument.

While making very accurate linguistic points, that article does not propose a formal framework in order to incorporate them. The attempts of implementation having got no further, it remains largely a theoretical issue. However, we have taken Nunberg's points into consideration for our own formulation.

2.2.3 Feature Logic for Dotted Types

One of the first independent formalizations for the Generative Lexicon, and its •-types specifically, has been proposed by [Pinkal and Kolhase, 2000]. Unfortunately, there was little follow-up and exchanges between the authors of this proposal and the persons looking for a complete formalization of the Generative Lexicon. The proposed *feature logic* had some promise, establishing a complete, Montague-compatible logic for compositional semantics that treated the information contained by complex types in records (a recent proposal, [Cooper, 2007], recently re-visited this prospect). Yet the logic was problematic in a fundamental way, in that any accommodation resulted in the loss of information, and thus, changes were untraceable.

2.2.4 Commonsense metaphysics and complex types

Well aware of the logical shortcomings of the original Generative Lexicon Theory, Pustejovsky set to describe a formally sound theory of types, together with Asher. The first effort, [Pustejovsky and Asher, 2000], has been met with skepticism, and the system is still the object of current work – including Chapter 5 of [Asher, 2008]. The purpose of this work is to model very specific constructions using solely the interaction of types, and to provide an accurate description of difficult cases, such as different quantification and individuation conditions for the separate, yet related aspects of a single, complex item.

Yet, the system remains largely unsatisfactory. The notations used are sometimes inconsistent, and the very axioms of the logic developed assume the availability of non-trivial operations on variables. The confusion of levels within the model creates a fundamentally flawed formalism.

2.2.5 The meeting of two theories

In [Marlet, 2007], the author attempts a straightforward gap-bridging between Computational Semantics and the Generative Lexicon. This proposal is firmly grounded and provides a solid implementation of two GL-supported operations, type coercion and selective binding, in compositional semantics. The calculus is quite complete; however, it remains a preliminary work, as many lexical operations are left unstudied, and as it presupposes the existence of working implementations of compositional semantics and of a Generative Lexicon.

2.2.6 A return to ontologies

Finally, [Saba, 2007] argues for a different approach altogether. While acknowledging the need for background information on the lexicon in order to process an utterance, and the pertinence of rich, ontology-based data structures akin to the Generative Lexicon to store and access this information, the author advocates that such a framework cannot be constructed *ex nihilo*. Instead, he proposes that the system should be used in order to *learn* the background lexical information automatically from corpora.

While making interesting points and presenting an approach that we partially concur with, this proposal is however too thin on known logical issues to be used as-is.

2.3 Our principles

The approach taken in this article, contrasting with the above methodologies, is to build incrementally upon a known, reliable framework: simply-typed λ-calculus *à la* Montague; however, we nevertheless wish to abide by the spirit of the Generative Lexicon as a linguistic theory. From our point of view, every addition, every aspect of lexical semantics integrated in that framework should be kept as simple as possible, logically sound, and linguistically motivated. More specifically, the compositional logic should keep track of changes induced by the lexicon, and be able to account for the variety of phenomena encompassed by polysemy.

These principles have led us to introduce the notion of *specific morphisms* in publications such as [Mery et al., 2007a] and [Mery et al., 2007b]. We model the potential semantic variations induced by lexical information using specific morphisms, represented by λ-terms, rather than canonical morphisms deduced from the typing system. This enables the framework to keep the relations between the different senses of a same word as distinct and diverse as need be, and to restrict the scope of each sense using specific constraints.

2.4 Target phenomena

In addition to the aforementioned *qualia exploitation* and *grinding* operations, we aim at modeling the co-predicative phenomena on certain kinds of words, represented and described as *dot objects* in [Pustejovsky, 2005]. Overall, our goals include :

- **Standard behaviour**: being able to account for classical composition and predication in phrases such as *a small stone*.
- **Qualia exploitation**: Modeling adequately the shift of sense occurring in the use of *qualia*, including *selective binding*. `Formal` exploitation is seen in the phrase *sunflower-yellow*, `agentive` exploitation in *a loving smile*, `constitutive` exploitation in *a powerful computer*, and `telic` exploitation in *a fair trial*, *an enjoyable cigarette* or *an easy book*.
- Modeling the similar shifts of senses occurring in *grinding* and in the use of *alternative aspects* for complex objects, in phrases such as *good-tasting salmon* or *a heavy book*.
- Including the different kinds of *complex objects*, as detailed in [Pustejovsky, 2005] :
  - Words encompassing both a `speech act` and a `proposition`, such as *promise*.
  - Words encompassing both a `state` and a `proposition`, such as *belief*.
  - Words encompassing both an `attribute` and a `value`, such as *height*.
  - Words encompassing both an `event` and some intrinsically bound content, such as *seminar*, *appointment*, *concert* or *dinner*.
  - Words encompassing both a `physical` object and an `informational` content, such as *book*, *file*, or *cd*.
  - Words encompassing both a `process` and its `result`, such as *reference*, *agreement* or *donation*.
  - Words encompassing both a `physical` object and an `aperture`, such as *door*.
  - Words encompassing a variety of associated aspects, such as *town*, which can refer to a *geographical locus*, a *group of people*, an *institution*, or more.
- **Complex type introduction**: accounting for the phenomenon which enables words with simple senses to be considered as *dot objects*, in phrases such as *reading the subway wall*.
- **Correct / Incorrect co-predication**: distinguishing between the cases where *co-predication* can be used, as in *a heavy but interesting book*, and where it may not, as in *?? delicious yet late-blooming lemongrass*.

3 Model outline

The details of our formalization must thus indicate how the model handles type coercion and similar constructs with specific morphisms, and integrate them to classical compositional semantics. We define the structures needed, and the notion of application needed in order to allow the use of these morphisms.

3.1 Architecture of the lexicon

The lexicon is rather organised as a type theoretic grammar à la Montague, which maps a lexical entry onto a finite number of λ-terms. More specifically, each lexeme yields a pair consisting of, on the one hand, a single term that we will refer to as the *main* λ-term, and, on the other hand, a finite, possibly empty set of terms that we will refer to as *optional terms* or *optional morphisms*.

The main λ-term ought to be used exactly once, and is like the usual term that Montague semantics associates with a lexical item.

The optional λ-terms are added to enable some composition that, without them, would be impossible; they can be used as many times as needed, including none. Anticipating slightly, let us assume that we have to apply a main term from the lexicon $\tau^{X\to Y}$ to a another such term u^{Z} with $Z \neq X$ yielding $(\tau\ u)$. If one of the two entries provides an optional morphism $i^{Z\to X}$ it becomes possible to insert this morphism at the right place, yielding $(\tau\ (i\,u)) : Y$. Observe that a morphism $j^{(X\to Y)\to(Z\to Y)}$ would do as well, yielding $((j\ \tau)\ u) : Y$.

The ontology, conceived as a set of inheritance relations, is then represented as a set of morphisms written $\iota^{X\to Y}$. If there are finitely many categories in the ontology there is a morphism $\iota^{X\to Y}$ whenever Y is an immediate superset of X. However, one could envision having such a morphism as soon as Y is a superset of X, but a more refined version would allow these morphims to be weighted, thus providing a measure of the distance of specialisation. If there are limit points, any subset of the actual inclusions which generates all of them by transitivity is fine.

3.2 Lambda calculus: types, terms, reduction

Since second-order λ-calculus is not very different from simply typed calculus, it might be convenient to use second-order types and terms, also known as polymorphic terms and types. More precisely, we use the second order propositional logic and second order λ-calculus as our type system. The second order propositional quantifier (as in Girard's system F) will be denoted by Λ (Girard's original notation). The second-order λ-abstraction $\Lambda\alpha.\ T$, viewed as a logical formula, would be written $\forall\alpha\ .T$, where the $\forall$ quantifies over propositions or types (*not* first-order variables). Thus, to say that a term has $\Lambda\alpha.\ T$ for type means that, for any arbitrary type U, it admits $T[U/\alpha]$ for type (the application $(\Lambda\alpha.\ T)\{U\}$ results in $T[U/\alpha]$).

From a typing system point of view, such a second-order type is a generic type, such as the ones used by (Ca)ML (yet more general, in the sense that this quantification can be nested in the right-hand side of implications or arrows). For instance, the type $\Lambda\alpha.\ \alpha\to\alpha$ is inhabited by the term $\Lambda\alpha.\ \lambda x.\ x$, i.e, the generic identity function, first applied to a type U and resulting in the identity $\lambda x^{U}.\ x$ of this type U.

Together with the ordinary, Montagovian constants e and t, we have a denumerable set P of type variables. To sum up:

- Constants e and t, as well as any type variable α in P, are types.
- Whenever T is a type and α a type variable which may but need not occur in T, $\Lambda\alpha.\ T$ is a type.
- Whenever T_1 and T_2 are types, $T_1\to T_2$ is also a type.

For each type, we have a denumerable set of variables of this type.

- A variable x of type T is a *term* of type T.
- Whenever τ is a term of type T and f is a term of type $T \to U$, $(f\ \tau)$ is a term of type U.
- Whenever x^T is a variable of type T, and τ a term of type U, $\lambda x^T.\ \tau$ is a term of type $T \to U$.
- Whenever τ is a term of type $\Lambda\alpha.\ T$, and U is a type, $\tau\{U\}$ is a term of type $T[U/\alpha]$.
- Whenever α is a type variable, and τ is a term of type T without any free variable involving the type variable α, $\Lambda\alpha.\tau$ is a term of type $\Lambda\alpha.T$.

The reduction is defined as follows:

- $(\Lambda\alpha.\tau)\{U\}$ reduces to $\tau[U/\alpha]$ (remember that α and U are types).
- $(\lambda x.\tau)u$ reduces to $\tau[u/x]$ (usual reduction).

As with the usual simply typed λ-calculus (but with much more sophisticate arguments), this system is strongly normalising and confluent (hence every term has a unique normal form): see [Girard, 1972].

3.3 Application revisited

Application takes place as usual when types match. In the event of a type clash, however, the second order logic is used to anticipate the transformation from a yet unknown type into a type appearing in the term.

Lexical items are provided with transformation terms that model specific morphisms, to solve type clashes. There are two ways of resolving such a pattern: globally, as in (1) below, and locally (i.e., independently, in specific places), as in (2) below. The local variant is of course more general, since the global solution might be viewed as the same local type transformation applying everywhere.

Given the problematic application situation:

$$(\lambda x^V.\ (P^{V\to W}x))\tau^U$$

The type clash might be resolved using either (1) or (2), depending on the available terms:

1. Global use of a transformation:

$$(\lambda x^V.\ (P^{V\to W}x))\ (f^{U\to V}\tau^U)$$

Assuming that f is an optional term associated with either P or τ. The transformation is applied directly to the argument, no matter the structure of the predicate – i.e., there might be several occurrences of the variable x, and each would result in a similar type clash, should the transformation f be unavailable. For instance, a conjunction would be resolved as $(\lambda x^V.\ (\wedge\ (P^{V\to W}x)\ (Q^{V\to W}x))\ (f^{U\to V}\tau^U)$, so that each occurrence of the argument would be transformed in the same manner.
Here, the type of the argument – and, therefore, the domain type of the transformation – is known. We could also write the resulting term $\Lambda\alpha.(\lambda x^\alpha.\ (P^{\alpha\to W}x))\ (f^{U\to\alpha}\tau^U)\{V\}$, in the manner of (2) below, but the second-order abstraction would be quite redundant here.

2. Local use of a transformation:

$$(\Lambda\alpha\lambda f^{\alpha\to V}.(\lambda x^{\alpha}.\ (P^{V\to W}(f^{\alpha\to V}x^{\alpha}))))\{U\}\ f^{U\to V}\ \tau^{U}$$

Again, assuming f is available to P or τ; this application variant actually infers the type $\{U\}$ and the associated morphism f from the original formula $(\lambda x^{V}.\ (P^{V\to W}x))\tau^{U}$. This construction provides "slots" for the local application of transformations.
Here, the type of the argument x is not known, and the term used to represent the predicate needs to capture it in order to infer the domain of the transformation f. Thus, the second-order abstraction is not an option in this construction.

3.4 Examples

– When there is no need to use additional lexical information, the application takes place as usual: this is **standard behaviour**. Here, φ is the type of physical objects:

small stone

$$\overbrace{\overbrace{(\lambda x^{\varphi}.\ (\text{small}^{\varphi\to\varphi}x))}^{\text{small}}\ \tau^{\varphi}}^{\text{stone}}$$
$$(\text{small}\ \tau)^{\varphi}$$

– In **qualia exploitation**, an information such as the presence of an `agentive` quale of type `person`, associated to the lexical entry for *smile*, will be modeled using a term $f_a^{S\to P}$ (to be used either globally or locally, and associated to the type S of *smiles*), that denotes the relationship between a smile and its author. Then, that transformation term will be used to model the appropriate type coercion:

wondering, loving smile

$$\overbrace{\overbrace{(\lambda x^{P}.\ (\text{and}^{t\to(t\to t)}\ (\text{wondering}^{P\to t}\ x)\ (\text{loving}^{P\to t}\ x)))}^{\text{wondering, loving}}\ \overbrace{\tau^{S}}^{\text{smile}}}^{}$$
$$(\text{and}^{t\to(t\to t)}\ (\text{wondering}^{P\to t}\ x)\ (\text{loving}^{P\to t}\ x)))(f_a^{S\to P}\tau^{S})$$
$$(\text{and}\ (\text{loving}\ (f_a\ \tau))\ (\text{loving}\ (f_a\ \tau)))$$

– **Incorrect co-predication**: destructive lexical operations, such as *grinding*, are modeled using transformation terms that can only be used globally. Thus, there is no correct way to derive, using a transformation term such as $f_g^{Fs\to Fd}$ that models the process of turning a fish into food, the infelicitous sentence below:

(??) *The tuna we had yesterday was lightning fast and delicious.*

– **Correct co-predication**: the relation between a word and its compatible aspects can be modeled using terms that might apply locally. Thus, let $f_p^{T\to P}$ and $f_l^{T\to L}$ represent the relations binding the words denoting a *town* to the associated *people* and *locus*, respectively; then the following co-predicative sentence is valid:

København is both a seaport and a cosmopolitan capital.

Intuitively, there should be a conjunction between predicates $cospl^{P\to t}$, $\text{cap}^{T\to t}$ and $port^{L\to t}$, applied to the same k^T with (where T stands for town, P for people, L for locus, k for København). If $T = P = L = e$, as in standard Montague semantics, then one would obtain $(\lambda x^e(\text{and}^{t\to(t\to t)}((\text{and}^{t\to(t\to t)}\ (\text{cospl}\ x)\ (\text{cap}\ x))\ (\text{port}\ x)))\ k$. Here, the canonical solution for building a well-typed term out of the *and*, the main λ-terms and the optional ones is the following:

If *and* is the usual constant of type $t\to(t\to t)$, the *and* between two predicates $P^{\alpha\to t}$ and $Q^{\beta\to t}$ with different domains α and β is:

$$\Lambda\alpha\Lambda\beta\lambda P^{\alpha\to t}\lambda Q^{\beta\to t}\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}.(\text{and}\ (P\ (f\ x))(Q\ (g\ x)))$$

This term first binds second-order variables to the domain types of P and Q, which are arbitrary α and β ($\Lambda\alpha\Lambda\beta$). The type of the argument x, also arbitrary, is then bound to ξ ($\Lambda\xi$). In order to apply that argument to both predicates, we need to provide two separate slots for independent transformations of the argument, according to the domain type of each predicate, and the conjunction thus looks like $(\text{and}\ (P\ (f\ x))(Q\ (g\ x)))$ (both transformations are local). We straightforwardly have the necessary typing for each transformation ($\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}$), and thus the above term. Remark that, in order to obtain a well-formed, well-typed term which can be reduced to a logical formula, two suitable morphisms have to be available and substituted to the transformation slots f and g.

The conjunction is first applied to P and T and to $cospl^{P\to t}$ and $cap^{T\to t}$, yielding:

$$(*)\qquad \Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to P}\lambda g^{\xi\to T}(\text{and}\ (\text{cospl}^{P\to t}\ (f\ x))\ (\text{cap}^{L\to t}\ (g\ x)))$$

Likewise, applying the same conjunction to $(*)$ and $(\text{port}\ x)$, we obtain:

$$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to P}\lambda g^{\xi\to T}\lambda h^{\xi\to L}(\text{and}\ (\text{and}\ (\text{cospl}^{P\to t}\ (f\ x))\ (\text{cap}^{T\to t}\ (g\ x)))\ (\text{port}^{L\to t}\ (h\ x)))$$

This term is then applied to T, the type of the argument, k^T and the morphisms. $cap^{T\to t}$ is already of the type of the argument, so we use $id^{T\to T}$. The result is a term of type t:

$$(\text{and}^{t\to(t\to t)}\ (\text{and}^{t\to(t\to t)}\ (\text{cospl}\ (f_p\ k^T)^P)^t)\ (\text{cap}\ (\text{id}\ k^T)^T)^t)^t\ (\text{port}\ (f_l\ k^T)^L)^t)^t$$

4 Properties of the model, advantages and drawbacks

4.1 Relationship with classical semantics and additional power

In order to fall back on classical compositional semantics, it is sufficient to consider that the lexicon simply provides the Montagovian typing of the terms. The additional terms and second-order types would just be discarded.

From a comparison with such a classical formalism, it is clear that our model does not increase the generative power of the semantics. Indeed, if anything, the generative power is less than with Montagovian semantics, as more unfelicitous or nonsensical sentences can be filtered out.

The benefits of this approach are of *expression*, as the output can be fine-grained by the choice of the morphism to use: thus, the fact that we refer to the *taste*, in the phrase *a good meal*, and to the *fitness for use*, in the phrase *a good tool*, is possible in our model, and made explicit by the use of terms specific to each relation supposed, without having to rely on the pragmatics or interpretation of the sentence in context.

4.2 Keeping track

In order to cope with co-predicative sentences, the system needs to be aware of the different changes of sense applied to an item. This is one of the main logical problems of the original type coercion, and of formalizations, such as [Pinkal and Kolhase, 2000] that attempt to implement it directly: if the lexical data licence it, the type is simply substituted. This means that co-predication can either be licensed in every or in no case. It also is problematic for anaphora resolution, and for any construction that uses the type-coerced term more than once.

This is why our formalism provides a straightforward way of keeping track of the changes used by explicitly inserting corresponding terms in the logical form.

4.3 Predicate-induced transformations

In [Pustejovsky and Asher, 2000], a puzzling situation (that led the authors to introduce numerous mechanisms and operations) appears in phrases such as *reading the subway wall*: **complex type introduction**. As the theory assumes that *read* selects for arguments objects with both `physical` and `informational` aspects aggregated in a complex type, there is no immediate and simple solution to that situation.

Our model, however, has a simple way of dealing with that problem: it is to suppose that the predicate, as well as the argument, might contribute transformation terms. Here, *read* conveys a term such as $f_a^{A\to\varphi\times I}$, converting artifacts into readable objects. This consideration obviates the need for specific operations, and does not, in fact, necessitate complex types at all.

4.4 Unresolved points

We have intentionally remained vague about a certain part of our proposal, namely: when does a transformation term apply locally, and when does it apply globally?

In fact, the different ways of applying the transformation terms is the means that the system uses to license or forbid co-predications. For some cases, there are clear answers: destructive operations such as grinding may only apply globally, while accessing the multiple tropes of a single item should be able to apply locally. However, on closer inspection, there are constraints that can prevent transformations from applying locally. Some syntactic constructions might be more acceptable than others (compare *a blue and open door* and *the blue door is open*), contrastive focus also changes the felicity of some constructions (compare *heavy and interesting book* and *heavy, yet interesting book*), etc.

As we have yet to determine the precise behaviour of co-predicative constructions and are not aware of existing studies to that effect, the choice between the two modes of application is left unspecified in our system.

5 Information, beyond the lexicon

With our model presented and supposed sound, we examine and speculate over some points of interest that might, with further work, be part of its applications.

5.1 Multiple possibilities

A straightforward extension of the formalism is to accept that application might behave in more than one way, i.e., that there might be more than one morphism that could be selected to attain the target type. By introducing such non-determinism in the derivation process, we would keep every possibility, and get effectively several different, yet potentially all valid in context, interpretations of the same sentence. For instance, *Philadelphia wants a bridge* offers several interpretations (*Philadelphia* can stand for the mayor, city council, inhabitants as polled, or some other city representatives).

But it would be useful to integrate a limited inference system, in order to cull inconsistent choices from the interpretation set as they appear. If one says *Philadelphia wants a bridge, but the mayor opposes it*, then the derivation branch corresponding to *mayor* would have to be discarded.

5.2 Scoring Felicity

The possibility of having multiple interpretations should be moderated by the fact that some interpretations are more likely to be correct than others. Likewise, some uses of words might feel a bit forced, a bit less felicitous than others, while remaining correct. Ideally, one would need to use some felicity "score" to respond to the classical objection presented, for instance, by [Blutner, 2002]: Putsejovsky argues that *want* has a tendency to select the `telic` quale, and, thus, that *want a cigarette* is taken to mean *want to smoke a cigarette*. But to *want a car*, conceivably, means to *want to possess a car*. Rather than using Blutner's philosophy (i.e., this distinction lies entirely within the pragmatics), simply allowing the interpretation to be multiple (i.e., for *want* to select either the telic or the object as a possession) suffices.

But we might allow the interpretations to have different scores, based on the relative value of the object, for instance (one might conceivably want Napoleon's cigarette for a possession, and a cheap car for driving).

5.3 Integrating multiple sources of prior knowledge

Beyond the knowledge intrinsic to words in the lexicon, other sources might play an analogous role, and be treated in the same way: as means to integrate specific morphisms over lexemes. The information associated to some words might evolve in the broad context of a *discourse*, for instance: generic sentences acquire different meanings if used with different topics. Thus, DRT, λ-DRT or SDRT are prime candidates to become additional information sources. Likewise, cultural assumptions might infer different meanings; there are many European texts that rely upon the presence of a church in any village, for instance. Some factors induced from pragmatics, such as the circumstances of the utterance and non-verbal signs, might add some modifiers as well. It is even envisionable, in such a theory, that the two speakers in a given dialogue do not have the same background knowledge of the lexicon, culture and circumstances, which could help model misunderstandings.

5.4 Acquiring new information from corpora

The use of Lexical Semantics – or, at least, of semantics of composition grounded in an adequate ontological structure – in order to provide a framework for automated knowledge acquisition from corpora is the main point of [Saba, 2007]. Though we do not entirely agree with the overall philosophy, the approach could be put to good use in order to facilitate the improvement of an existing lexicon grounded on the above principles. The idea is simple: if a sentence is grammatically correct, but not valid from the point of view of the Generative Lexicon because the lexicon is incomplete (i.e., some specific morphisms remain yet unknown or undiscovered), then the program will consider that there must needs exist some morphisms that allow the sentence to be validated. The missing morphisms are added to the lexicon as underspecified type-conversions, and as much information as possible is gathered on their uses as they occur in the text; they can be either acknowledged and corrected, or discarded, by a human judge on a later phase.

5.5 A link with knowledge representations

In retrospect, the use of *background knowledge* used for the parsing of sentences goes back to [Searle, 1979], yet Pustejovsky's lexicalization of the phenomenon, and our formalization, restrict the scope of Searle's original hypothesis. It remains true that we are lead to distinguish between two aspects of world knowledge, in a theory of the language speaker's mind: the knowledge needed to process a sentence, the background, which has been the object of our model, and the knowledge obtained as a result of the processing of that sentence, the facts. We would like to speculate, at this point, that an adequate representation of this duality might successfully be combined with a many-worlds approach.

World models (incorporating both the background knowledge and resulting facts) would then be constructed for every agent and every sentence or discourse. The confrontation of this world with the agent's knowledge of its reality (that is also a world model) constructs the interpretation of the sentence.

Such a view of the interpretation of language is largely compatible with the Lexical Semantics theorized before, and should be familiar to multi-agent advocates. Indeed, it resembles some of the approaches recently developed to analyse dialogue, and it is our opinion that it would give interesting and applicable results, when further investigated.

Conclusion

While the above research goals could in time lead to many possibilities, we think that the proposed mechanisms are now sufficient to integrate many facets of the Generative Lexicon Theory, while remaining formally sound. In order to be of actual use, however, they need to be provided with a translation strategy from lexical entries to transformation terms. This would be a first step towards an actual implementation of the formalism, and its evaluation.

Acknowledgements We would like to thank the many people that commented and discussed our ideas, previous publications, addresses and drafts, including every member of the Signes group, attendants and members of the audience at ESSLLI 2007, and Nicholas Asher, for their inputs. Thanks go to Reinhard Muskens for making the NDTTG workshop possible, and for inviting us to write this article.

References

[Asher, 2006] Asher, N. (2006). A Type Driven Theory of Predication with Complex Types. Unpublished MS.

[Asher, 2008] Asher, N. (2008). *A Web of Words: Lexical Meaning in context*. Unpublished MS.

[Blutner, 2002] Blutner, R. (2002). Lexical Semantics and Pragmatics. *Linguistische Berichte*.

[Cooper, 2007] Cooper, R. (2007). Copredication, dynamic generalized quantification and lexical innovation by coercion. In *Fourth International Workshop on Generative Approaches to the Lexicon*.

[Girard, 1971] Girard, J. Y. (1971). Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types. In Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symp.*, pages 63–92, North Holland, Amsterdam.

[Girard, 1972] Girard, J. Y. (1972). Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur. Thèse de Doctorat d'État, Université Paris VII.

[Girard et al., 1989] Girard, J.-Y., Taylor, P., and Lafont, Y. (1989). *Proofs and types*. Cambridge University Press, New York, NY, USA.

[Gupta and Aha, 2003] Gupta, K. M. and Aha, D. M. (2003). Nominal Concept Representation in Sublanguage Ontologies. In *Second International Workshop on Generative Approaches to the Lexicon*.

[Gupta and Aha, 2005] Gupta, K. M. and Aha, D. W. (2005). Interpreting Events Using Generative Sublanguage Ontologies. In *Third International Workshop on Generative Approaches to the Lexicon*.

[Jacquey, 2004] Jacquey, E. (2004). Ambiguïté lexicale et quantification : une modélisation de la polysémie logique. In Corblin, F. and Gardent, C., editors, *Interpréter en contexte*, pages 107–141. Hermès Science Publications, Paris.

[Marlet, 2007] Marlet, R. (2007). When the Generative Lexicon meets Computational Semantics. In *Fourth International Workshop on Generative Approaches to the Lexicon*.

[Mery et al., 2007a] Mery, B., Bassac, C., and Retoré, C. (2007a). A montagovian generative lexicon. In *Formal Grammar*.

[Mery et al., 2007b] Mery, B., Bassac, C., and Retoré, C. (2007b). A montague-based model of generative lexical semantics. In Muskens, R., editor, *New Directions in Type Theoretic Grammars*. ESSLLI, Foundation of Logic, Language and Information.

[Nunberg, 1993] Nunberg, G. (1993). Transfers of meaning. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 191–192, Morristown, NJ, USA. Association for Computational Linguistics.

[Pinkal and Kolhase, 2000] Pinkal, M. and Kolhase, M. (2000). Feature Logic for Dotted Types: A Formalism for Complex Word Meanings. In *ACL 2000*.

[Pustejovsky, 1995] Pustejovsky, J. (1995). *The Generative Lexicon*. MIT Press.

[Pustejovsky, 2005] Pustejovsky, J. (2005). A Survey of Dot Objects. Author's weblog.

[Pustejovsky and Asher, 2000] Pustejovsky, J. and Asher, N. (2000). The Metaphysics of Words in Context. *Objectual attitudes, Linguistics and Philosophy*, 23:141–183.

[Saba, 2007] Saba, W. S. (2007). Compositional Semantics Grounded in Commonsense Metaphysics. In *EPIA 2007*.

[Searle, 1979] Searle, J. (1979). *Expression and Meaning*. Cambridge University Press.