Typed Hilbert Operators for the Lexical Semantics of Singular and Plural Determiner Phrases

Bruno Mery, , Christian Retoré

Typed Hilbert Operators for the Lexical Semantics of Singular and Plural Determiner Phrases —Short Abstract—

Bruno Mery[1], Richard Moot[1], Christian Retoré[2]

IUT de Bordeaux, Université de Bordeaux, France
LaBRI — CNRS
bruno.mery@me.com, richard.moot@labri.fr
Université de Montpellier, France
LIRMM — CNRS
christian.retore@lirmm.fr

Abstract

We have proposed a framework based upon the λ-calculus with higher-order intuitionistic types for the symbolic computation of the semantic analysis, integrating lexical data ([Bassac et al., 2010, Mery, 2011, Retoré, 2014]). We discuss the pertinence of using Russell's ι and Hilbert's ε and τ operators for the semantics of the definite and indefinite determiners, recapitulating the main points of [Retoré, 2014], extending the work of von Heusinger in [Egli and von Heusinger, 1995] with higher-order tyiping in a multi-sorted frameowrk. We then study the possible implications of using such operators on underspecified sets of individuals such as those used when computing the semantics of plurals or massive entities, as done in [Mery et al., 2015].

1 Overview

The original idea beneath the selection operators is Russell's Iota, which is $\iota_x.F$: the unique individual x such that $F(x)$. It was later amended by von Heusinger: a contextually salient x such that $F(x)$ (asserting unicity is notoriously problematic for logics that include negation). It is intuitively suitable for modelling the semantics of definite determiners such as *the*.

Hilbert provided two related operators. The first is Epsilon, in which $\varepsilon_x.F$ provides an existential quantification for x, which exists whenever F is satisfiable. This operator is suitable to model the semantics of indefinite determiners such as *a*. The second is Epsilon's dual Tau, written $\tau_x.F$, a form of universal quantification for x in F (it is true that $F(\tau_x.F)$ when all possible individuals x satisfy F).

They denote the selected individuals : $\exists x.F(x)$ is $F(\varepsilon_x.F)$, $\forall x.F(x)$ gets written as $F(\tau_x.F)$, *the cat* as $\iota_x.\lambda x.(\mathrm{cat}\ x)$ (and $\mathrm{cat}(\iota_x.\lambda x.(\mathrm{cat}\ x))$ is true).

The operators ε and τ being dual ($\varepsilon_x.F = \tau_x.\neg F$), ε is sufficient and can be used to (with standard application and reduction) to form a logical calculus. In its original formulation, the Epsilon calculus is a conservative extension to first-order logic, as the classical $\forall, \exists$ quantifications can be translated straightforwardly using ε, τ (sometimes yielding overly complicated formulae). However, some formulae of the Epsilon calculus are not equivalent to any first-order formula (such as $P(\varepsilon_x.Q)$).

In [Egli and von Heusinger, 1995], von Heusinger uses ε and η to model determiners such as *the*, or *a*. He insists that there are no fundamental differences in semantics between the operators, η being used to introduce new referents. In this work, the quantification does not have the same strength as classical existential quantification ($\varepsilon_x.F$ is a term, but does not necessarily entail $F(\varepsilon_x.F)$). He demonstrated the pertinence of Hilbert's operators for the semantics of natural language, as a possible alternative to choice (Skolem) functions (see also [von Heusinger, 2004]).

2 Types for Determiners in NTT

Lexical semantics imply a system of types with many sorts and second-order operators, in the view of so-called New Type Theories (NTT). There have been several efforts to model the theories introduced in [Pustejovsky, 1995], including [Asher, 2011] and our own ΛTY_n (details in [Retoré, 2014]). As our goal is to implement the resulting system in an operational syntax/semantics parser (Grail, see [Moot, 2010]), we need to integrate many complex phenomena in a single, sound logical framework and keep computer tractability. We have thus been interested in variations of the Epsilon calculus for determiners. See [Retoré, 2014].

There are many different opinions, in a post-Montagovian, New Type Theory world, as to what constitutes a correct semantics for determiners. The full scope of those issues is far too broad for a single article, but they include co-variance and contra-variance in sub-typing (see e.g. [Luo et al., 2013]). For those reasons, in [Luo, 2012], Zhaohui Luo's option is to have common noun as types. Then determiners should be typed with $\Pi\alpha.\alpha \to \alpha$: the constant representing a is the term $\Lambda\alpha\lambda x^{\alpha}.x$ that, given a nominal type, yields an individual of that type (this amounts to a semantic identity, but can be interpreted in different ways).

We, however, adhere to the Montagovian view of common noun as predicates (modelling the property of being an individual of such denomination); this is much more convenient for quantificational purposes. Then determiners should be typed with $\Pi\alpha.(\alpha \to \mathbf{t}) \to \alpha$. The corresponding term would yield an individual (different according to the context and the interpretation of the specific determiner) that enjoys the property of the predicate passed as argument. This view is motivated by our approach of the granularity of the type system: we favour a system with many sorts corresponding to salient cognitive distinctions. *Human* should be a distinct type, *Child* could be a distinct type, but *Broccoli-liking-children* should most probably not be a type, yet we need to be able to parse DPs such as *the children who like broccoli*. We argue that the easiest way to do so is to have common nouns be predicates over a finite (and relatively small) number of basic types, and that *children who like broccoli* is the predicate $\lambda x^{H}.\text{children}^{H\to\mathbf{t}}(x) \wedge \text{like\_broccoli}^{H\to\mathbf{t}}$ over the type H of human agents. We develop this typing for determiners in Section 3.

Note that, if need be, we can also model the alternate view with this typing by having $\hat{\alpha}$ a predicate of type $\mathbf{e} \to \mathbf{t}$ for any type α. This fits neatly with von Heusinger's vision of Hilbert's operators.

3 Typing the Operators for Use in our System

We use ε, η mostly as von Heusinger does, using ε to signify "a (new) individual in the context of enunciation that satisfies the property" and to model the indefinite determiner *a* (as Hilbert's original notation and von Heusinger's η) and ε^{1} to signify "the most salient/interesting existing individual in context that satisfies the property" and to model the "definite" determiner *the* (as von Heusinger's η, and Russel's ι minus the unicity condition). These operators have identical operational semantics, different interpretations in context ; we add an existential presupposition $(F(\varepsilon_x.F), F(\varepsilon^{1}{}_x.F)$ respectively) in every case, as speaking about something asserts its essence. Those presuppositions, similar to Asher's in [Asher, 2011] can reside with typing judgements and contextually salient entities and facts in a dedicated area that can be implemented via Cooper storage, Luo's signature structure, multi-dimensional analysis, or simply conjoined to the main formula – for the sake of simplicity, we will not include the resulting presuppositions in the typing of the operators, as it is pretty straightforward but dependent on the chosen paradigm. As before, τ is the dual of ε (and also entails $F(\varepsilon_x.F)$).

Our principal contribution is to provide a typing in ΛTY_n that enables the operators to be used in our framework, and to have a polysemous interpretation depending on context. That typing is $\varepsilon : \Pi\alpha.(\alpha \to \mathbf{t}) \to \alpha$, a single polymorphic operator (again, ε^{ι} has the same type and semantic term with a different interpretation). When applied to a predicate $F(x)^{\mathbf{t}} = \lambda x^{a}.(F\ x)$ for some type a, the operator is specialised for a.

(For the sake of simplicity of the resulting formulae, we will use the usual $\forall, \exists$ quantification for variable-binding along with $\varepsilon, \varepsilon^{\iota}$).

4 Groups and Sets: Using Hilbert's Operators with Predicate Variables

We model groups and massive entities using specific operators, with predicates denoting sets, and an operator $|\_|$ giving the cardinality of the different individuals satisfying a predicate (we hold that utterances only predicate on a finite number of distinct individuals) . We have specific sorts for group individuals and mass individuals, and lexical transformations ensure that everything goes smoothly; see [Mery et al., 2015] for details. *A team* is $\varepsilon_x.\lambda_x{}^{\mathbf{g}_H}.(\text{team } x)$. *Someone in the team* is $\varepsilon_x.\lambda x^H.\text{member\_of}(x, (\varepsilon^{\iota}{}_y.\lambda y^{\mathbf{g}_H}.(\text{team } y)))$. This follows Link's classical account of plurals and masses found in [Link, 1983], adapted for multiple sorts.

The "easy" way to model *Some people in the team* would simply to select all pertinent individuals and aggregate them together in a group. We have detailed a straightforward way to account for set-theoretic union in [Mery et al., 2013] based on Link's account, constructing a predicate representing the set of all individuals. However, as constructing a predicate to represent the set of individuals implies quantification over predicates, we could directly apply the ε operator to the target predicate and have much more control over the result. We can simply use $\varepsilon_P.\lambda P^{H\to\mathbf{t}}.(R^{(H\to\mathbf{t})\to\mathbf{t}}\ P)$, with the Epsilon operator quantifying over the set of individuals, (represented as their characteristic predicate) and applied to the desired property. The full semantics of *Some people in the team* would then be

$$\varepsilon_P.\lambda P^{H\to\mathbf{t}}.(|P| > 1 \wedge (\forall x^H.P(x) \Rightarrow \text{member\_of}(x, \varepsilon^{\iota}{}_y.(\lambda y^{\mathbf{g}_H}.(\text{team } y)))))$$

Allowing ε and ε^{ι} to be specialised for any type gives us sufficient flexibility to quantify over predicates, relations and sets. As we already use quantification over predicates (and types) as features of our system (*John and Mary* is modelled as $\lambda x^H.(x = j) \vee (x = m)$), we should not refrain from doing so with determiner quantification.

Conclusion

We believe that typed, polymorphic Hilbert-derived operators are suitable tools to model the various polysemous usages of determiners. Those operators avoid the standard pitfalls of simplistic quantification such as symmetry (*some A are B*) and incomplete (nominal) sentences. We have shown that they can be adapted for recent frameworks including lexical semantics and contextual adaptation by the means of an higher-order typing system. The handling of their interpretation, and the correct usage of presuppositions and their inferences/implicatures remains to be investigated, for instance when combining quantification over a set of individuals and distinct anaphoric references to some of those individuals. We are confident that those issues can be resolved within our system.

References

[Asher, 2011] Asher, N. (2011). *Lexical Meaning in Context: a Web of Words*. Cambridge University Press.

[Bassac et al., 2010] Bassac, C., Mery, B., and Retoré, C. (2010). Towards a Type-Theoretical Account of Lexical Semantics. *Journal of Language, Logic, and Information*, 19(2).

[Egli and von Heusinger, 1995] Egli, U. and von Heusinger, K. (1995). The epsilon operator and e-types pronouns. *Lexical Knowledge in the Organization of Language*, pages 121–141.

[Link, 1983] Link, G. (1983). The logical analysis of plurals and mass terms: A lattice-theoretic approach. In Portner, P. and Partee, B. H., editors, *Formal Semantics - the Essential Readings*, pages 127–147. Blackwell.

[Luo, 2012] Luo, Z. (2012). Common nouns as types. In Béchet and Dikovsky, editor, *Logical Aspects of Computational Linguistics*, pages 173–185. Springer.

[Luo et al., 2013] Luo, Z., Soloviev, S., and Xue, T. (2013). Coercive subtyping: Theory and implementation. *Inf. Comput.*, 223:18–42.

[Mery, 2011] Mery, B. (2011). *Modélisation de la Sémantique Lexicale dans le cadre de la Théorie des Types*. PhD thesis, Université de Bordeaux.

[Mery et al., 2013] Mery, B., Moot, R., and Retoré, C. (2013). Plurals: individuals and sets in a richly typed semantics. In Yatabe, S., editor, *LENSL'10 - 10th Workshop on Logic and Engineering of Natural Semantics of Language, Japanese Symposium for Artifitial Intelligence, International Society for AI - 2013*, pages 143–156, Hiyoshi, Kanagawa, Japan. jSAI-ISAI, Keio University.

[Mery et al., 2015] Mery, B., Moot, R., and Retoré, C. (2015). Computing the Semantics of Plurals and Massive Entities using Many-Sorted Types. In *To appear in the selected proceedings of LENLS 11*, Lecture Notes in Computer Science. Springer.

[Moot, 2010] Moot, R. (2010). Wide-coverage French syntax and semantics using Grail. In *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Montreal.

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

[Retoré, 2014] Retoré, C. (2014). The Montagovian Generative Lexicon Lambda Ty_n: a Type Theoretical Framework for Natural Language Semantics. In Matthes, R. and Schubert, A., editors, *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 202–229, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Retoré, 2014] Retoré, C. (2014). Typed Hilbert Epsilon Operators and the Semantics of Determiner Phrases. In *Formal Grammar 2014*, Tübingen, Germany.

[von Heusinger, 2004] von Heusinger, K. (2004). Choice functions and the anaphoric semantics of definite nps. *Research on Language and Computation*, 2:309–329.