Formalisation of Intensionality as Algorithms

Roussanka Loukanova
Computational Linguistics, Uppsala University and TiLPS, Tilburg Universiy
rloukano@stp.lingfil.uu.se

1 Background and Recent Development

In a sequence of papers, Moschovakis developed a class of languages of recursion as a new approach to the mathematical notion of algorithm and development of computational semantics, e.g., see Moschovakis [7], for FLR, and Moschovakis [8], for L^{λ}_{ar}. In particular, the language and theory of acyclic recursion L^{λ}_{ar} is intended for modeling the logical concepts of meaning and synonymy, from the perspective of the theory of computability, by targeting adequateness of computational semantics of NL. L^{λ}_{ar} is a higher order type theory, which is a proper extension of Gallin's TY_2, Gallin [3], and, thus, of Montague's Intensional Logic (IL). L^{λ}_{ar} has a highly expressive language, an effective reduction calculus and strong mathematical properties. It models the notion of algorithm by abstract mathematical objects, which are tuple of functions defined by mutual recursion, called acyclic recursors. The referential intensions of the meaningful L^{λ}_{ar} terms are acyclic recursors defined by their canonical forms, which are *recursion terms*. For the construction of recursion terms (where-terms), the language L^{λ}_{ar} uses a recursion operator, denoted by the constant where that applies over a *head* term A_0 and a set of assignments, called *body*, $\{p_1 := A_1, \ldots, p_n := A_n\}$, where each A_i is a term of the same type as the recursion variable p_i $(1 \leq i \leq n)$: A_0 where$\{p_1 := A_1, \ldots, p_n := A_n\}$. The where-terms represent recursive computations by designating functional recursors: intuitively, the denotation of the term A_0 depends on the functions denoted by $p_1, \ldots, p_n$ that are computed recursively by the system of assignments $\{p_1 := A_1, \ldots, p_n := A_n\}$. In an acyclic system of assignments, the computations close-off. The formal syntax of L^{λ}_{ar} allows only recursion terms with acyclic systems of assignments, while the FLR allows cyclicity, but is limited with respect to its type system. The languages of recursion (e.g., FLR and L^{λ}_{ar}) have two semantic layers: denotational semantics and ref-

Proceedings of the 8th International Conference on Computational Semantics, pages 295–298,
Tilburg, January 2009.

erential intensions. The recursion terms of L_{ar}^{λ} are essential for encoding that two-fold semantic information. **Denotational Semantics:** For any given semantic structure $\mathfrak{A}$, there is at most one, well-defined denotation function, den, from terms and variable assignments to objects in the domain of $\mathfrak{A}$. Thus, for any variable assignment g, an L_{ar}^{λ} term A of type σ *denotes* a uniquely defined object $\mathsf{den}(A)(g)$ of the subdomain $\mathfrak{A}_{\sigma}$ of $\mathfrak{A}$. L_{ar}^{λ} has a reduction calculus that reduces each term A to its canonical form $\mathsf{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}$ (unique modulo congruence, i.e., with respect to renaming bound variables and reordering of assignments). **Intensional Semantics:** The notion of intension in the languages of recursion covers the most essential, computational aspect of the concept of meaning. Intuitively, $\mathsf{Int}(A)$ is the *algorithm* for computing its denotation $\mathsf{den}(A)$. Formally, the *referential intension*, $\mathsf{Int}(A)$, of a meaningful expression A is the recursor that is defined by the canonical form $\mathsf{cf}(A)$ of A. Two meaningful expressions are synonymous iff their referential intensions are naturally isomorphic, i.e., they are the same algorithms. Thus, the algorithmic meaning of a well-formed expression (i.e., its sense) is the information for how to "compute" its denotation, i.e., expressions have sense by carrying instructions for acquiring what they denote in a structure (model). The canonical form $\mathsf{cf}(A)$ of a meaningful term A encodes its intension, i.e., the algorithm for computing its denotation, via: (1) the basic semantic facts, which consist of $\{p_1 := A_1, \ldots, p_n := A_n\}$ and the "head pattern" A_0, that are needed for computing the denotational interpretation $\mathsf{den}(A)$, and (2) a rank order of the steps for incremental computation of the denotation $\mathsf{den}(A)(g)$, e.g., a terminating order of the recursive steps that compute each $\mathsf{den}(A_i)(g)$, for $i \in \{0, \ldots, n\}$. Thus, the languages of recursion offer a formalisation of central computational aspects of Frege's distinction between sense and denotation, with two semantic "levels":

$$\underbrace{\mathsf{NL\ Syntax} \Longrightarrow\ L_r^{\lambda} \Longrightarrow \mathsf{Referential\ Intensions\ (Algorithms)} \Longrightarrow \mathsf{Denotations}}_{\mathsf{ComputationalSemantics}}$$

2 Open Problems for the Language of Acyclic Recursion as Semantic Theory of NL

Relational Type Theory with Partiality Acyclic recursion terms and acyclic recursors model terminating algorithms for computing the denotations of meaningful expressions. The idea of restricting recursion languages to acyclicity, as in L_{ar}^{λ}, is that, for certain applications to NL semantics, partiality and self-reference are not needed. In 80's, Barwise and Perry

(e.g., [1]) introduced Situation Theory with the ideas that partiality, factual content and situatedness are crucial features of the meaning concepts that involve mental states, incl. attitudes. Situation Theory models partiality and the inherent relational and situational nature of information, in general, not only linguistic, by diverging from the traditional type theoretic settings. Situation Semantics proceeds as a special case of application of Situation Theory to NL semantic information. By taking up the ideas of partiality, Muskens [9] realized the ideas of Situation Semantics by generalizing Gallin's TY_2 with partial relations and building corresponding generalized Montague grammars, i.e., Partial Type-theoretic Grammars. The importance of partial relational structures to semantics of NL is well investigated by Situation Theory. Furthermore, Muskens [9] demonstrated that encoding relational type systems is not only inadequate in the case of partial relational structures, but needless. That opens a need of extending the language L^{λ}_{ar} to a full higher order type theory for modeling recursors with partial functions and relations.

Factuality and State Variation L^{λ}_{ar} uses *states* (similar to indexes for possible worlds and times, situations, contexts) at all levels of its own syntax and semantics. However, the potential expressiveness of L^{λ}_{ar} for representation of state dependant semantic objects has not been fully developed. There is a need of more finely grained semantic concepts by a type theory of recursion that: (a) represents denotation functions with values that are partial, situated objects with factual content, and, (b) uses terms with internal variation of state constants and state variables that occur inside terms. Such terms are more adequate representation of information that depends on varying[1] states. Work in the direction of representing locality of semantic facts in L^{λ}_{ar}, has been done (however without state variation in individual terms) by Kalyvianaki [5],[4].

Denotation and Intention Muskens [10] refined the denotation function by a revision of Thomason's Intentional Logic. The result is a logical grammar that takes propositions as primitive objects, with a relation that associates propositions with sets of possible worlds. The techniques of Muskens [10] offer a possibility for refining the denotational semantics of recursion languages by splitting it into two sub-layers: (a) *situated denotations* (as above), and, (b) *denotational intention*:

$$\mathsf{Syn}\ L^{\lambda}_{r} \Longrightarrow \mathsf{Intensions\ (Algorithms)} \Longrightarrow \mathsf{Denotations} \begin{cases} \text{Sit Denotations} \\ \text{Denotational Intentions} \end{cases}$$

[1] Similarly to, for example, the varying *resource situations* in Situation Theory.

Underspecification Underspecified semantic representation became major effort of contemporary research, see Bunt [2] for a comprehensive overview of the field. Representing semantic underspecification of NL with languages of recursion needs to be developed. Initial work, see Loukanova [6], shows the unique expressiveness of L^{λ}_{ar} to accommodate its inherent facilities for representing semantic underspecification of NL expressions.

Representation of Attitudes Semantics of attitudes, such as *know*, *believe*, etc., in the languages of recursion, is an open problem.

Syntax-Semantics Interface A major work to be done is to define *render* relations from NL to languages of recursion. Initial work shows that that is a realistic task. A more demanding task, with potential applications, e.g., to machine translation, is a render relation with inverse to NL.

References

[1] J. Barwise and J. Perry. *Situations and Attitudes*. Cambridge, MA:MIT press, 1983.

[2] H. Bunt. Semantic underspecification: Which technique for what purpose? In H. Bunt and R. Muskens, editors, *Computing Meaning*, volume 3 of *Studies in Linguistics and Philosophy 83*, pages 55–85. Springer, Dordrecht, 2007.

[3] D. Gallin. *Intensional and Higher-Order Modal Logic*. North-Holland, 1975.

[4] E. Kalyvianaki. *Algorithmic Natural Language Semantics*. PhD thesis, University of Athens, 2007.

[5] E. Kalyvianaki. Factual content in algorithmic natural language semantics. In V. V. Nurmi and D. Sustretov, editors, *Proceedings of the Twelfth ESSLLI 2007 Student Session*, pages 123–133, Dublin, Ireland, 2007.

[6] R. Loukanova. Typed lambda language of acyclic recursion and scope underspecification. In R. Muskens, editor, *Workshop on New Directions in Type-theoretic Grammars*, ESSLLI 2007, pages 73–89, Dublin, Ireland, 2007.

[7] Y. N. Moschovakis. Sense and denotation as algorithm and value. Number 2 in Lecture Notes in Logic, pages 210–249. Springer, 1994.

[8] Y. N. Moschovakis. A logical calculus of meaning and synonymy. *Linguistics and Philosophy*, 29:27–89, 2006.

[9] R. Muskens. *Meaning and Partiality*. Studies in Logic, Language and Information. Stanford: CSLI Publications, 1995.

[10] R. Muskens. Sense and the computation of reference. *Linguistics and Philosophy*, 28:473–504, 2005.