Type-Theoretical Semantics with Coercive Subtyping*

Zhaohui Luo
Dept of Computer Science,
Royal Holloway, Univ of London
Egham, Surrey TW20 0EX, U.K.
zhaohui@cs.rhul.ac.uk

Summary There have been a lot of interesting developments in lexical semantics including, for example, the Generative Lexicon Theory [9]. However, the research so far has failed to provide a satisfactory formal account to explain the important linguistic phenomena in the lexical theories. Most of the employed formalisms are based on (extensions of) Montague grammar [7] and unable to capture the linguistic phenomena satisfactorily. This paper studies type-theoretical semantics and shows that the modern type theory, together with the theory of coercive subtyping [5], offers a powerful and adequate logical language for formal semantics in which interesting lexical phenomena such as logical polysemy [9] and copredication [2] can be properly interpreted.

Type Theory and Coercive Subtyping Some of the basic ideas of studying logical semantics in type theory have been considered by Ranta based on Martin-Löf's type theory [10] where, in particular, common nouns are interpreted as types (rather than as functional subsets of entities as in Montague grammar). This is natural in type theory, but has some unwelcome consequences since there are fewer operations on types as compared with those on functional subsets. As shown in the paper, subtyping is not only useful but crucial in solving this problem and coercive subtyping provides us with such a framework.

Coercive subtyping [5] is a general theory of subtyping for dependent type theories such as Martin-Löf's type theory [8] and UTT [4]. The basic idea is to consider subtyping as an abbreviation mechanism: A is a subtype of B ($A \leq B$) if there is a unique implicit coercion c from type A to type B and, if so, an object a of type A can be used in any context $\mathcal{C}_B[\_]$ that expects an object of type B: $\mathcal{C}_B[a]$ is legal (well-typed) and equal to $\mathcal{C}_B[c(a)]$. For a type theory with nice meta-theoretic properties such as Strong Normalisation (and hence logical consistency), its extension with coercive subtyping has those properties, too. In computer science, coercive subtyping has been implemented in many proof assistants such as Coq, Lego, Matita and Plastic, and used effectively in interactive theorem proving. As shown in this paper, when applied to linguistic semantics, coercive subtyping plays a crucial role in application of type theory to logical semantics.

Coercive Subtyping in Type-Theoretical Semantics In a type-theoretical semantics, common nouns are interpreted as types and verbs and adjectives as predicates. For example, we have $[\![\texttt{book}]\!], [\![\texttt{human}]\!] : Type$, $[\![\texttt{heavy}]\!] : [\![\texttt{book}]\!] \rightarrow Prop$ and $[\![\texttt{read}]\!] : [\![\texttt{human}]\!] \rightarrow [\![\texttt{book}]\!] \rightarrow Prop$, where $Prop$ is the type of propositions. Modified common noun phrases can be interpreted by means of Σ-types of dependent pairs: for instance, $[\![\texttt{heavy book}]\!] = \Sigma([\![\texttt{book}]\!], [\![\texttt{heavy}]\!])$.

Now, how could we reflect the fact that, for example, a `heavy book` is a `book`? Such phenomena are captured by means of coercive subtyping, by declaring the first projection π_1 as a coercion: $\Sigma(A, B) \leq_{\pi_1} A$. For example, if $h : [\![\texttt{human}]\!]$ and $b : [\![\texttt{heavy book}]\!]$, then $[\![\texttt{read}]\!](h, b)$,

*Abstract for presentation at Semantics and Linguistic Theory 20 (SALT20), Vancouver, 2010.

the interpretation of 'h reads b', is well-typed (by coercive subtyping). This shows that coercive subtyping solves a key problem in type-theoretical semantics. This problem is discussed in [10] (pp. 62-64), where three possible solutions are considered, but none of them is completely satisfactory. One of them is closest to ours where explicit first projections are employed; it is one step short: using π_1 as an implicit coercion, we have managed to capture the phenomena as intended.

Subtyping relations propagate through the type constructors such as Π and Σ (they become $\rightarrow$ and $\times$ in the non-dependent cases, respectively). For instance, they propagate through the function types, contravariantly: if $A' \leq A$ and $B \leq B'$, then $A \rightarrow B \leq A' \rightarrow B'$. For example, $[\![\texttt{book}]\!] \rightarrow Prop \leq [\![\texttt{heavy book}]\!] \rightarrow Prop$.

As another example, let's consider the dot-types as studied in [9]. Let $\textsc{Phy}$ and $\textsc{Info}$ be the types of physical objects and informational objects, respectively. One may consider the dot-type $\textsc{Phy} \bullet \textsc{Info}$ as the type of the objects with both physical and informational aspects. Intuitively, a dot-type is a subtype of its constituent types: $\textsc{Phy} \bullet \textsc{Info} \leq \textsc{Phy}$ and $\textsc{Phy} \bullet \textsc{Info} \leq \textsc{Info}$. A book may be considered as having both physical and informational aspects, reflected as: $[\![\texttt{book}]\!] \leq \textsc{Phy} \bullet \textsc{Info}$. By contravariance,

$$\textsc{Phy} \rightarrow Prop \leq \textsc{Phy} \bullet \textsc{Info} \rightarrow Prop \leq [\![\texttt{book}]\!] \rightarrow Prop$$
$$\textsc{Info} \rightarrow Prop \leq \textsc{Phy} \bullet \textsc{Info} \rightarrow Prop \leq [\![\texttt{book}]\!] \rightarrow Prop$$

Therefore, for example, for $[\![\texttt{burn}]\!] : \textsc{Phy} \rightarrow Prop$ and $[\![\texttt{interesting}]\!] : \textsc{Info} \rightarrow Prop$, '`burn an interesting book`' can be interpreted as intended.[1]

Dot-Types, Lexical Entries and Coercion Contexts In the type-theoretical semantics with coercive subtyping, several useful constructions can be defined and used to model various linguistic phenomena. They include (and the details will be in the full paper): (1) *Dot-types*: Although the meaning of a dot-type [9] is intuitively clear, its proper formal account has been surprisingly tricky (see, for example, [1]). In type theory with coercive subtyping, a dot-type $A \bullet B$ can be defined by means of the product type $A \times B$ *together with* its two projections as implicit coercions, provided that 'the components of A and B are disjoint'. This gives, for the first time to our knowledge, an adequate formal treatment of dot-types and can hence be used in a satisfactory way in formal semantics to interpret, for instance, *copredication* as discussed in [2] and logical polysemy [9]. (2) *Lexical entries* as studied in the Generative Lexicon Theory [9] can be expressed formally as dependent record types [6]. (3) *Coercion contexts* can be introduced to model the more complicated phenomena such as those involving reference transfers [3].

References

[1] N. Asher. A type driven theory of predication with complex types. *Fundamenta Infor.*, 84(2), 2008.

[2] N. Asher. *Lexical Meaning in Context: A Web of Words.* 2009. (Draft book to be published by CUP).

[3] R. Jackendoff. *The Architecture of the Language Faculty.* MIT, 1997.

[4] Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science.* OUP, 1994.

[5] Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1):105–130, 1999.

[6] Z. Luo. Manifest fields and module mechanisms in intensional type theory. In *LNCS 5497*, 2009.

[7] R. Montague. *Formal Philosophy.* Yale University Press, 1974. (Collection edited by R. Thomason).

[8] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory.* OUP, 1990.

[9] J. Pustejovsky. *The Generative Lexicon.* MIT, 1995.

[10] A. Ranta. *Type-Theoretical Grammar.* OUP, 1994.

[1] In Montague grammar (and its extensions), common nouns are interpreted as functional subsets of type $e_0 \rightarrow t$, where t is the type of propositions and e_0 is a subtype of the type of entities. For instance, $[\![\texttt{book}]\!] : \textsc{Phy} \bullet \textsc{Info} \rightarrow t$ and $[\![\texttt{heavy}]\!] : (\textsc{Phy} \rightarrow t) \rightarrow (\textsc{Phy} \rightarrow t)$. In such a situation, in order to interpret, e.g., '`a heavy book`', one would have to apply $[\![\texttt{heavy}]\!]$ to $[\![\texttt{book}]\!]$ by requiring, for example, $\textsc{Phy} \bullet \textsc{Info} \rightarrow t \leq \textsc{Phy} \rightarrow t$, which is not the case – type clashes would happen, leading to unnatural and complicated treatments [1].