An Analysis of Selectional Restrictions in Dependent Type Semantics
- Logical polysemies, Coercion and Metaphor -

Department of Computer Science   Eriko Kinoshita (Supervisor : Daisuke Bekki)

1. Introduction. Recently, various extended type systems have been developed to explain phenomena related to coercion [1, 2, 13]. These systems are enriched with more fine-grained types than is commonly assumed in formal semantics, which enables us to analyze selectional restrictions of predicates as type matching. In such systems, an operation for dealing with coercion is triggered when a predicate and its argument cause a type mismatch. However, there is a problem with the naive assumption that coercion is only triggered by type mismatch: coercion can be contextually triggered without any type mismatch. For example, (1a) has the reading in (1b), even though the literal reading does not produce a type mismatch.

(1) a. The lion escaped.
b. The actor who plays the lion at *The Lion King* escaped (from the theater).

Moreover, type inference with selectional restriction is context-sensitive in a way that is analogous to *presupposition* (see §3). These phenomena suggest that coercion is not only driven by type matching in semantic composition but also by inference with contextual information.

This paper proposes a formal analysis of coercion in the framework of Dependent Type Semantics (DTS) [3, 4], a framework of proof-theoretic semantics that combines dependent types with underspecification. Coercion is treated at the stage of type checking implemented as *proof search*, rather than at the stage of semantic composition. We show that this analysis captures the inferential aspect of coercion as shown by phenomena like (1).

2. DTS. Dependent types have been used to analyze various aspects of natural languages [6, 12]. Compared with simple types, dependent types can express *types depending on terms*. For instance, $\mathbf{man}(x)$ is a type depending on a term x that corresponds to the proposition that x is a man. Dependent types include Π-type (dependent function type), written $(x : A) \to B$, and Σ-type (dependent product type), written $(x : A) \times B$ or $\left[\begin{smallmatrix} x : A \\ B \end{smallmatrix}\right]$ (see [10] for details). A term having a dependent type is called a *proof term*. For instance, $p : \mathbf{man}(x)$ expresses that a proof term p has the type $\mathbf{man}(x)$, in other words, p is a proof for the proposition that x is a man.

In contrast to model-theoretic semantics, the semantics of natural language based on dependent types can be called *proof-theoretic semantics*, where the meaning of a sentence is regarded as a *proof-condition* (verification condition), rather than as a truth-condition. By taking proof-conditions as a central aspect of meaning, semantic theories based on dependent types are particularly suitable for capturing inferential aspects of interpretation.

DTS [3, 4] differs from the previous studies in that it has underspecified term @ to handle phenomena that depend on the preceding contexts such as presupposition and anaphora. For example, the sentence in (2) is given the semantic representation (SR) containing the term @ having the Σ-type for the proposition that there is an entity x such that x is a man. A term having the Σ-type is a pair of objects, from which the projection function π_1 takes the first element.

(2) He whistled.

$$\mathbf{whistle}\left(\pi_1\left(@ : \begin{bmatrix} x : \mathtt{entity} \\ \mathbf{man}(x) \end{bmatrix}\right)\right)$$

Such an underspecified SR is compositionally derived from a parse tree for a sentence. Then the well-formedness of an SR is proved via type-checking. This process of type-checking involves a process of constructing a proof term for @. By replacing @ with the constructed term, we can resolve presupposition and anaphora and thus obtain a fully specified SR for the sentence. The overall process is shown in (3). See [3, 4] for more details.

3. An Analysis in DTS. Selectional restrictions show the behavior of presupposition [1, 8, 9].

(3) $\boxed{\text{parse trees}} \xrightarrow{\text{semantic composition}} \boxed{\text{underspecified SRs}} \xrightarrow[\text{(proof search)}]{\text{type-checking}} \boxed{\text{well-formed SRs}}$

$$
\dfrac{\dfrac{}{\mathbf{escape} : \begin{bmatrix} x : \texttt{entity} \\ \mathbf{animate}(x) \end{bmatrix} \to \mathsf{type}}\,(CON) \qquad \dfrac{\dfrac{\dfrac{\overline{} \\ \vdots}{\mathcal{TF}^{\,l}_{\mathbf{animate}} : \mathsf{type}} \quad \dfrac{\vdots}{? : \mathcal{TF}^{\,l}_{\mathbf{animate}}}}{(@ : \mathcal{TF}^{\,l}_{\mathbf{animate}}) : \mathcal{TF}^{\,l}_{\mathbf{animate}}}\,@}{\pi_1(@ : \mathcal{TF}^{\,l}_{\mathbf{animate}}) : \begin{bmatrix} x : \texttt{entity} \\ \mathbf{animate}(x) \end{bmatrix}}\,(\Sigma E)}{\mathbf{escape}(\pi_1(@ : \mathcal{TF}^{\,l}_{\mathbf{animate}})) : \mathsf{type}}\,(\Pi E)
$$

Figure 1: Type checking for the SR of "The lion escaped."

For instance, the predicate *marry* presupposes that its arguments are human; this presupposition projects out of constructions such as negation:

(4) Bob didn't marry Mary.
$\Rightarrow$ Bob and Mary are human.

To capture these presuppositional inferences, we analyze selectional restrictions as presuppositions in DTS (cf. [8]). For example, we define a lexical entry for *escape* as follows.

$$\textit{escape} : \lambda x.\mathbf{escape}(\pi_1(@ : \mathcal{TF}^{\,x}_{\mathbf{animate}}))$$

where $\mathcal{TF}^{\,x}_{\mathbf{animate}} \equiv$

$$\begin{bmatrix} t\_pair : \begin{bmatrix} y : \texttt{entity} \\ \mathbf{animate}(y) \end{bmatrix} \\ F\_source : \texttt{entity} \to \mathsf{type} \\ p : F\_source(x) \\ R : \begin{bmatrix} w : \texttt{entity} \\ F\_source(w) \end{bmatrix} \to \begin{bmatrix} y : \texttt{entity} \\ \mathbf{animate}(y) \end{bmatrix} \to \mathsf{type} \\ R(x,p)t\_pair \end{bmatrix}$$

In simple type theory, the type of a one-place predicate is usually defined as $\texttt{entity} \to \textit{prop}$. In DTS, by contrast, the type of $\mathbf{escape}$ is $\begin{bmatrix} x : \texttt{entity} \\ \mathbf{animate}(x) \end{bmatrix} \to \mathsf{type}$, where $\mathbf{animate}(x)$ specifies the selectional restriction of the predicate. Note that selectional restrictions are represented as *predicates*, not as types. In the case of (1b), we have an SR $\mathbf{escape}(\pi_1(@ : \mathcal{TF}^{\,l}_{\mathbf{animate}}))$, where l is the term of *the lion*. The type checking for this SR goes as shown in Fig. 1. Here the term @ launches a proof search for the missing information: it searches a 5-tuple $(t\_pair, F\_source, p, R, q)$ that satisfies the condition $\mathcal{TF}^{\,l}_{\mathbf{animate}}$ from the context, where q is a proof term for $R(l,p)t\_pair$. In short, @ acts to locate an object of type $\texttt{entity}$ that satisfies the property $\mathbf{animate}$, which is the selectional restriction of predicate *escape*, associated with the lion from the context. More specifically, t_pair is a pair (a,t) of an actor a (of type $\texttt{entity}$) and a proof t that the actor is animate; F_source is a property of the lion l such that the first argument of the relation R has F_target as a selectional restriction. In this case, we can choose the predicate "*y plays w at the Lion King*" for R and $\mathbf{animate}$ for F_target. By replacing @ with the constructed term and taking its first projection in terms of π_1, we can obtain an SR $\mathbf{escape}(a,t)$, which captures the intended reading of (1b). In the case of the literal reading of (1a), we can use the equality relation to fill in R, which leads to the desired interpretation. Using the mechanism of presupposition projection in DTS, we can also account for the inference patterns in (4).

Although details are omitted here due to space limitations, the present theory can naturally extend to other phenomena discussed in the literature [11], in particular, to logical polysemies (copredication) [1], metaphor and aspectual verbs [7]. In our proof-theoretic account, an effort to obtain an interpretation by coercion can be measured by the complexity of a proof search. This accounts for an aspect of interpretation involved in metaphor, coercion and aspectual verbs called *complicity* [5]. In the full version of the paper, a more detailed comparison with other type-theoretic approaches where coercion is treated at the stage of semantic composition [1, 2] will be addressed too.

References. [1] Asher, N. (2011) *Lexical Meaning in Context*.[2] Asher, N. and Luo, Z (2012) Formalisation of coercions in lexical semantics.[3] Bekki, D. (2014) Representing anaphora with dependent types. [4] Bekki, D. & Mineshima, K. (2017) Context-passing and underspecification in Dependent Type Semantics. [5] Camp, E. (2017) Why metaphors make good insults. [6] Cooper, R. (2012) Type theory and semantics in flux. [7] Egg, M. (2003) Beginning novels and finishing hamburgers. [8] Kinoshita, E., Mineshima, K. & Bekki, D. (2016) An analysis of selectional restrictions with Dependent Type Semantics. [9] Magidor, O. (2013) *Category mistakes*. [10] Martin-Löf, P. (1984) *Intuitionistic Type Theory*.[11] Pustejovsky, J:1995, The Generative Lexicon.[12] Ranta, A. (1994) *Type Theoretical Grammar*.[13] Retoré, C. (2014) The Montagovian Generative Lexicon ΛTy_n.