Formalization of Coercions in Lexical Semantics

Nicholas Asher (CNRS, France) & Zhaohui Luo (Royal Holloway, Univ of London, UK)

Originally developed for use in programming languages with simple type systems [4], coercions between various types, including many subtypes of the type of entities (eventualities, physical objects, informational objects, etc.), have been widely employed in semantics [6,5,7,1]. Coercions, however, pose a problem for lexical semantics: as detailed in [1], [7] fails to provide a formally explicit or empirically adequate analysis of coercion examples; and the only explicit model theory for coercions with subtyping [1] is category theoretic, which does not ensure a consistent logic. However, the approach in [1] is close to that in [2]; one can exploit [2]'s notion of coercive sub typing within the framework of [1]. A combination of [1] and [2] provides a consistent logic and a nice treatment of a lot of data as considered in [1], and leads to an analysis of new data that are not accounted for by any previous theory.

Coercive subtyping [2] is a theory of subtyping developed for modern type theories (MTTs) in their applications to proof development in verification of systems and formalisation of mathematics. It construes subtyping as an abbreviation mechanism: A is a subtype of B if there is a unique implicit coercion c from A to B, notation $A <_c B$. This simple idea when used with rich type structures in an MTT is powerful and very useful for formal semantics [3]: it subsumes both injective subtyping and projective subtyping, incorporates parameterized coercions, and provides a principled solution to the problem of subtyping. The coercive subtyping extension is conservative [8], with the consequence that the consistency of the basic theory is preserved.

The basic coercive subtyping mechanism that coerces $f(a)$ into $f(c(a))$ by inserting the coercion c into a gap between f and a, suffices to represent many linguistic coercions. For example, the type-shifting law in [6] that lifts the e-type of an NP to its GQ-type can be modelled with the coercion $c[A]$ parameterized by A of subtype $\textsc{cn}$ (the universe that consists of the types that interpret CNs): $A <_{c[A]} (A \to \mathit{Prop}) \to \mathit{Prop}$, where $c[A](x) = \lambda P : A \to \mathit{Prop}.\ P(x)$. It can be verified that these coercions commute with the other subtyping relations and hence satisfy the required coherence condition in coercive subtyping.

To extend the basic forms of coercive subtyping to treat event coercions along the lines discussed in [1], we introduce coercion contexts and local coercions. Consider *John started a novel* with $[\![start]\!] : \mathit{Human} \to \mathit{Event} \to \mathit{Prop}$; its formal interpretation $\exists x : [\![novel]\!] .\ [\![start]\!](j, x)$ is well-typed, because '$[\![start]\!](j, x)$' can be coerced to $[\![start]\!](j, [\![reading]\!](x))$. However, in such an example there are in fact context-dependent 'multiple coercions': e.g., the above sentence could have meant 'John started writing a novel'; there could also be several reading events of that novel. Since coercive subtyping requires contextual uniqueness of coercions (coherence), we must restrict the scope/context in the interpretations using *coercion contexts*, an extended formal notion of context in type theory with entries expressing coercion assumptions, and *local coercions* of the form **coercion** c **in** t, where the expression t is the scope of coercion c [3]. Considered in this way, much of [1]'s semantic ideas can be encoded with coercive subtyping.

Dependent types in MTTs can model some examples that require functors in [1]. For instance, instead of *Event*, we may consider dependent types $\mathit{Evt}(h)$, the type of events conducted by $h : \mathit{Human}$. But dependent types alone are not sufficient; we also need parametrized coercions to replicate the account in [1]. Assuming that *start*, *read* and *write* have type $\Pi h{:}\mathit{Human}.\ (\mathit{Evt}(h) \to \mathit{Prop})$ and considering parametrized coercions $\mathit{Book} <_{c(h)} \mathit{Evt}(h)$, where $c(h, b) = \mathit{write}(h, b)$, if h wrote b, and $c(h, b) = \mathit{read}(h, b)$, otherwise (simplifying the latter case for readability), the semantics of the first sentence of (d) can be given in (e(1)), which is coerced into (e(2)).

a. Jill just started a book. #It started at 10 am.

b. The omelet is eating himself/?? it/?? itself.

c. The omelet left without paying because he found it inedible.

d. Jill just started *War and Peace*, which Tolstoy finished in 1820. But it$_{\text{reading}}$ won't last; she never finishes any book she starts.

e. (1) $\mathit{start}(j, \mathit{wp}) \land \mathit{finish}(t, \mathit{wp})$ (2) $\mathit{start}(j, \mathit{read}(j, \mathit{wp})) \land \mathit{finish}(t, \mathit{write}(t, \mathit{wp}))$

f. $\exists e : \textsc{event}\ (\mathrm{start}(j, e) \land \mathrm{read}(e, j, \mathit{wp}) \land \exists e' : \textsc{event}\ (\mathrm{finish}(t, e') \land \mathrm{write}(e', t)))$

Scope controls as provided by means of local coercions [3] do not suffice to handle all event coercions. For example, consider (d) above: the coercion from *the book* to *reading the book* must be limited to the first clause so that the coercion from *the book* to *writing the book* may occur in the relative clause. But we need the "read" coercion to get the anaphoric reference to the event in the third clause. The framework in [1] provides a compositional treatment of coercions, where a type mismatch and an instruction on the type requirements of the predicate can introduce a functor transforming an argument's type to an appropriate one; e.g., the presence of a coercing predicate like *enjoy* licenses a functor transforming the predicational context into one that combines with *book*'s type and resolves the mismatch. The functor introduces an underspecified polymorphic event type whose exact value depends on the type of *book*, *enjoy*'s other arguments and further information in the discourse context. [1] exploits an internal semantics using types and an external, model theoretic semantics for standard semantic logical forms; the functor has a spell out in the logical form with the external semantics and produces an existentially bound variable for the (underspecified) eventuality that *enjoy* requires as its internal argument; this allows for multiple coercions without a requirement of uniqueness and provides the correct logical form for (d), when an appropriate context is provided. The logical form for the first clause of (d) is in (f).

A combination of spell out and scope control is required to predict the data in (a-d); they are both needed and must be treated as distinct operations. The spell out mechanisms in [1] alone incorrectly predict that the coerced eventuality in (a,b) is anaphorically available but correctly predict the felicity of (c,d), while the cope control mechanism as provided by local coercions, when used correctly, may lead to a semantic modelling method that seems to make right predictions in (a,b) but wrong predictions for (c,d). We can adapt the compositional process of [1] to use a scope controlled coercion that leads to spell out in appropriate circumstances. Details are in the full paper.

What controls the process of taking scope controlled coercion to spell out in the external semantics? We believe some notion non locality is at work. Roughly, non-locality requires that felicitous anaphoric reference to a coerced eventuality or other object must span two spatio-temporally disconnected events. Once the anaphor occurs within a non-local situation with respect to the coerced object spell-out occurs and produces a variable that is available for anaphoric binding. The semantics of the rhetorical connections between clauses sketched in [0] can help determine non-locality. The second clause of (a) is an elaboration or reformulation of the second; according to the semantics of such a relation, the second clause describes a sub event of the first mentioned event. The scope for local coercion is then limited to the first clause, spell out does not occur, and the pronoun cannot find an appropriate antecedent. On the other hand, (c,d) involve relations of explanation and contrast and two distinct facts or events; thus, spell out can occur and the anaphoric link is felicitous. We hypothesize that the presence of non locality, often signaled by discourse relations, triggers spell out. This provides a new way to think of coercion and to study the effect of discourse structure on local predications, something discussed in preliminary fashion in [1]. But the current account goes substantially further, accounting for data that neither [1] nor any other formally explicit account can handle.

References [0] N. Asher & A. Lascarides. Logics of Conversation. CUP. 2003. [1] N. Asher. Lexical Meaning in Context: A Web of Words. CUP, 2011. [2] Z. Luo. Coercive subtyping. J Logic & Computation 9(1). 1999. [3] Z. Luo. Formal Semantics in Modern Type Theories with Coercive Subtyping. Ling. & Phil., 2012. (to appear) [4] J. Mitchell. Coercion and type inference. POPL'83, 1983. [5] M. Meons and M. Steedman. Temporal ontology and temporal reference. Comput. Ling. 14(1). 1988. [6] B. Partee and M. Rooth. Generalised conjunction and type ambiguity. 1983. [7] J. Pustejovsky. The Generative Lexicon. MIT, 1995. [8] S. Soloviev and Z. Luo. Coercion completion and conservativity in coercive subtyping. APAL 113(1-3), 2002.