

Formalization of Coercions in Lexical Semantics

Nicholas Asher — CNRS, IRIT, France

Zhaohui Luo — Royal Holloway, Univ of London, UK

1. Introduction

Originally developed for use in programming languages with simple type systems (see, for example, [22]), coercions between types, including many subtypes of the type of entities (eventualities, physical objects, informational objects, etc.), have been widely employed in linguistic semantics [26, 30, 28, 3]. There are, however, two problems with coercions. First, it has been unclear how coercions should be treated in formal semantics. There have been several unsatisfactory proposals (or even ‘half-proposals’ in that they do not provide an adequate formal framework) including, for example, [30], and a rich and formal framework is called for to provide adequate treatments. Secondly, coercions pose another problem for lexical semantics: as discussed in [3], most of the traditional set-theoretic models for type structures in linguistics fail to provide a coherent treatment of subtyping, while the only explicit model theory for coercions with subtyping proposed in [3] is category theoretic and does not necessarily provide a consistent logic. [14, 17] show that modern type theories (MTTs) enriched by coercive subtyping provide a promising proof-theoretic framework for an adequate treatment of subtyping and coercions. Some model-theoretic ideas considered in [3] can be reflected proof-theoretically in the MTT-approach. For instance, as shown for the first time in this paper, some data that require sophisticated coercion mechanisms can be given proper semantics by dependent types and the associated coercive subtyping mechanisms. In this paper, we review these two approaches and introduce some new data that pose interesting problems, requiring a refinement of these approaches. In particular, we show that many coercions are subject to discourse structural constraints in order for their coerced content to be made fully explicit.

2. Linguistic Coercions

In *coercion* a predicate whose standard selectional restrictions are not met by its argument may still convey a meaning because either one of the terms or the predication relation between predicate and argument is adjusted in some way so that the composition process may succeed. Sometimes this adjustment is needed to handle phenomena like coordination as in

- (1) Pat and at least three classmates went skiing together.

If *Pat* denotes an individual, it cannot combine with the generalized quantifier *at least three classmates* in the coordinate construction. [26] argue for a type shifting mechanism from expressions of type E to $(E \rightarrow T) \rightarrow T$ to adjust the types in (1) so that the complex DP can be constructed. Since that time, linguists have developed a rich and complex typology of types or sorts that are subtypes of E to handle a wide range of phenomena in semantics. Consider

- (2) a. John brought a bottle. It had a nice label/ It was yummy.
 b. John brought a bottle. It had a nice label and was yummy.

Depending on the continuation, one could infer that the first sentence of (2) with makes available a discourse referent for the bottle or its contents that can be linked to the anaphoric pronoun in the continuation; but as (2b) shows, the first sentence makes available discourse referents for *both* the bottle and its contents.

There are many other examples of coercion, for example those involving aspectual verbs like *start*, *begin* and *finish*, as well as verbs like *enjoy* in English:¹ (3) is equivalent in meaning to (4):

- (3) Julie enjoyed (started/finished) a book.
 (4) Julie enjoyed doing something with (e.g., reading, writing, ...) a book.

enjoy requires an event as its direct object as in *enjoy the spectacle*, *enjoy the view*. This also happens when *enjoy* takes a question as its complement, as in *enjoy (hearing) what he said*. When the direct object of a transitive use of *enjoy* does not denote an event, it is “coerced” to denote some sort of eventuality.

To extend type shifting mechanisms to such coercions, we must do several things. First, we must extend the type system “downwards” to include appropriate subtypes of E; we have to have subtypes for CONTAINER and CONTENTS and INDIVIDUAL PHYSICAL OBJECTS and EVENTUALITIES as inputs to the type shifting mechanism, whatever form this may take. Second, if we take, as we do, coercions to be licensed by conflicts between the requirements of predicates like *be yummy* or *have a nice label*, we have to assign these predicates types or lexical meanings such that they require one of these types as arguments. If we try to extend the standard system [8] in Montague Grammar where subtype or \sqsubseteq is understood as subset,² atomic types are understood as sets and predicates are understood as functions from the objects of the domain type to truth values, we run into a severe problem. We cannot extend the subtyping hierarchy from the atomic types to the predicates in any obvious way.³

Within the standard view of subtyping there are three options for handling the phenomena with type coercion, and only one of them, we think, is viable. The first possibility is to assume that the mechanism of type coercion transforms denotation of the noun phrase *the book* into some sort of eventuality denoting expression?⁴ If that is the case, then how can we access in subsequent discourse the referent of *the book*?⁵

- (5) Julie enjoyed a book. It was a mystery.

¹These have been discussed at length in the linguistics literature, in particular by [29].

²This is the so-call *subsumptive subtyping*, with the subsumption rule: if *A* is a subtype of *B*, then every object of type *A* is of type *B*.

³For more details see [3].

⁴The account in [29] seems to adopt this line of attack toward the problem.

⁵This problem makes the qualia story as developed in [29] a non starter as argued in [3].

These observations are familiar but show that we *cannot* shift the meaning of *the book* to some sort of eventuality. Or at least if this happens, whatever process is responsible for the shift must also allow *the book* to retain its original, lexical meaning and its original contribution to logical form.

The second alternative is to shift the predicate. In (3) *enjoy* should mean something like *enjoyed doing something with*.⁶ At the very least, however, one will need some underspecified form of coercion to handle cases of gapping coordination like the one below.

(6) Julie enjoyed (reading) a book and Isabel (watching) a movie.

Anaphora tests using ellipsis are instructive. The meaning of the activity is predicted on the predicate modification view to shift with the choice direct object during semantic composition. But if that is the case, then the ellipsis predicts a peculiar reading of the second clause of (7a) where Isabel enjoyed doing something to or with the parade, when it doesn't have that: Isabel simply enjoyed the parade as an event. Or perhaps she enjoyed being in the parade. There is a sense that Isabel and Julie did different things with the book and the parade—and shifting the predicate doesn't do justice to this intuition.

- (7) a. Julie enjoyed her book and Isabel her parade.
 b. Julie enjoyed a book and watching a movie.
 c. I really enjoyed the book and how it uses someone's life in the story and gives ideas on how to stay unstressed and calm. (<http://www.silverhillsontheroad.com/teens/teenbook>)

(7b-c) spell trouble for the predicate modification view. To handle the DP coordination, the predicate modification view predicts that Julie enjoyed doing something to watching a movie, which is uninterpretable. All this is strong evidence that coercion does not involve a *semantic* adjustment to the predicate at the moment of composition.

Since we've seen that you can't simply shift the meaning of the predicate and you can't shift the meaning of the argument and you can't relegate the problem of coercion to the pragmatics garbage can, what is left? One answer (see the section on TCL) is that you change the relation of predication that holds between the predicate and the argument. A clash between the type of the argument and the type presupposition of the predicate induces not a type shift in either the argument or the predicate but rather a type shift on the predication relation itself, which is implemented by introducing a functor that is inserted around the argument. Thus, the meaning of the argument does not shift—it remains what it always was; the predicate also retains its original meaning. What changes is how they combine. This account makes the right predictions for (7).

Coercions are also not just dependent on the predicate and the argument that “undergoes coercion” [3]. It may also depend on other arguments of the verb.⁷

(8) The painters have started the windows (meaning to paint the windows)

⁶[25], and others have pursued this line of attack.

⁷And on the discourse context generally.

- (9) The cleaners have started the suits but won't finish them before tomorrow (cleaning the suits)

So the coercion must take into account in principle *all* of the coercing predicate's arguments.

In addition, coercions affect the predication locally to the verb. This contrasts with the behavior of predications involving so called *dual aspect nouns* like *book* or *lunch*. These nouns are called dual aspect nouns because they appear to have simultaneously two incompatible types—ABSTRACT OBJECT and PHYSICAL OBJECT for *book* and FOOD and EVT for *lunch* [3]. When a verb selects one of their types, we note a change in the quantificational domain of the DP. Consider the following examples from [3].

- (10) a. John has mastered every mathematics book in the library. (Here the quantification is over abstract objects. John has mastered the contents, and may never have been to the library)
- b. John has stolen every mathematics book in the library (Here the quantification has to be over physical volumes; John has to have stolen say all the copies of Rudin's *Real and Complex Variable Analysis*).
- c. *The best of Robert Louis Stevenson* is three great books in one (here we have a quantification over abstract objects in the first DP and over physical volumes in the DP with the N' ellipsis).

Shifts in the interpretation of the DP, however, don't occur with most coercions. Consider

- (11) George enjoyed/started many books last weekend.

In (11), the quantification over books remains a quantification over books and not over eventualities. That is, (11) does not have a reading on which there are multiple events of George's enjoying just one book, which should be possible if coercions licensed quantificational shifts in the way that dual aspect nouns do. Thus, whatever coercion does, it makes its adjustments locally to the verb in logical form and does not affect the DP interpretation or quantificational domain.

Finally, coercions are language dependent. (12), a sentence considered in (3), is for example not good in Chinese.

- (12) Julie started a book.

In Chinese, one has to say (13) instead:

- (13) Julie started reading/writing a book.

The coercion that inserts 'reading/writing' actions is not licensed in Chinese when 'start' is used.

Chinese is also different from English even with respect to robust coercions like the container/contents one. For instance, consider the first sentence of (2) in Chinese. There are three ways to express (2) in Chinese: we can use BOTTLE₁ (Ping in Chinese), which is a classifier, or BOTTLE₂ (Ping-Zi in Chinese) which means a container called bottle:

- (14) John brought a BOTTLE₁.
 (15) John brought a BOTTLE₁ of wine.
 (16) John brought a BOTTLE₂.

One of the possible (context dependent) meanings of (14) is (15), where ‘a BOTTLE₁’ is coerced into ‘a BOTTLE₁ of wine’ and that (16) means that John bought a container. As we have seen in English we can follow *John brought a bottle* either by referring to the contents (17) or the container (18). But in Chinese (17) is good following (14) or (15), but (18) is not; and, following (16), (18) is good, but not (17).

- (17) It was yummy.
 (18) It had a nice label.

In order to express (18) following (14) or (15) in Chinese, one would have to say something like ‘The bottle had a nice label.’

The above phenomenon makes coercion a semantic phenomenon, not a pragmatic one. Pragmatic accounts use principles that are supposed to follow from general principles of rational interaction between agents; they are expected to be universal. So we should expect pragmatic enrichment to allow coercions, say from objects to eventualities, whenever the grammar allows for an eventuality reading in the argument. But this isn’t true.

We see two plausible options for dealing with this problem, options that we have detailed in print in [3] and [14, 17]. We sketch the details of both views here and consider some further developments.

3. Coercions in Type Composition Logic

Developed in [3],⁸ Type Composition Logic (TCL) integrates on top of a standard intensional semantics for English a system of types with a category theoretic or proof theoretic interpretation. In this system, subtyping is a matter of limited deduction. A lexical entry in this system thus conveys two levels of meaning, one considering the type requirements on entry’s arguments or its selectional restrictions (if any) and a consequent typing of the term itself, and a standard denotational meaning. Types are used to guide the construction of logical form. When a predicate requires that an argument have a type τ , then as long as its argument type σ is compatible with τ (i.e. $\sigma \sqcap \tau \neq \perp$), type checking succeeds and the predicate and the argument combine together in TCL with the argument taking as type the meet \sqcap of σ and τ .

With coercions like (2) or (3), however, there is no common sub type for the type presupposition of the predicate and the argument; the contents of a bottle is of a type incompatible with that of the bottle that contains it, and similarly, the eventuality required by *start* or *enjoy* is a distinct and

⁸but with precursors in [2].

incompatible type with that of a physical or abstract object like a book—these types have different identity and individuation criteria.

However, predicates like *start* license a justification of their type requirements by allowing one to introduce material that links the actual argument to an object of the desired type in a manner akin to the sort of presupposition justification employed in bridging. This changes the relation between the predicate and its arguments: the predicate allows a natural transformation from a given type to the desired type. This transformation is translated into the language of logical forms and has a truth conditional content. So in particular, for example (3), the transformation licenses the introduction of a functor that links the verb and the original argument and introduces into logic a description of an eventuality. But this description is itself underspecified, because without a particular context, we don't know exactly what Julie did to the book that she enjoyed. The logical form then that we want to end up with is this:

$$(19) \quad \exists x \exists e (book(x) \wedge enjoy(j, e) \wedge \phi(j, x, e))$$

where $\phi(j, x, e)$ is the underspecified description of the eventuality e . Note that e is now the argument of *enjoy*, as is required, but that the discourse entity introduced by *a book* is also available for anaphoric reference. Note further that the meaning of *book* remains what it always was; it's just that it combines in a different way with the verb from the usual function application method.

We now specify the TCL lexical entry for *enjoy* (and *mutatis mutandis* other event coercing predicates), taking into account the discussion of the last section. In TCL a transitive verb takes two generalized quantifier arguments. To pass type requirements of the verb to the arguments, TCL uses the method of continuations; the typing context is made part of the lexical entry and is labelled by the lambda bound variable π . π is a list of type requirements provided by the context; a verb simply adds its type requirements to the list. Coercion verbs justify apply their type presuppositions locally, namely in the predication of the verbal predicate to its arguments.

$$(20) \quad \lambda \Psi \lambda \Phi \lambda \pi \Phi(\pi * AG) \lambda v \Psi(\pi) \\ (\lambda y_1 \lambda \pi_1 (\text{enjoy}(v, y_1, \pi_1 * ARG_2^{\text{enjoy}} : \text{EVT} - \epsilon(\text{HD}(\Phi), \text{HD}(\Psi))))))$$

This says that the subject DP must contribute an agent argument of type AG to the first argument position of *enjoy* and that the object DP contribute a second argument of eventuality or EVT type.⁹ The fact that *enjoy* licenses a coercion for its direct object argument is signalled in the type requirements associated with this section argument. In effect the type requirement on the object is complex; *enjoy* requires that its second argument be an eventuality but it allows a natural transformation from the head type of the DP (the type of objects that the DP quantifies over or denotes) to an underspecified type that is a function of this head type and of the head type of the subject DP. Nevertheless, the term itself does not have the complex type associated with a function from agents to eventualities to propositions (or truth values); it has the type of being a function from

⁹*enjoy* is a control verb and imposes that the agent of the eventuality is the subject of *enjoy*, but we skip this detail here.

general DP denotations to DP denotations to typing contexts to propositions. This dissociation between the type requirements of the verb's arguments and the type of the term allows us to apply type justifications and adjustments in the predication relation at various points in the composition process, allowing us to distinguish between composition involving dual aspect nouns and coercion.

To see how to derive the logical form in (19), let us apply this lexical entry to the arguments it is supplied with in (3). We assume that *Julie* has the usual entry for a name in MG, except that it specifies its denotation as an agent. The DP *a book* has the entry:

$$(21) \quad \lambda P \lambda \pi \exists x (book(x, \pi * \text{BOOK}) \wedge P(x)(\pi * \text{BOOK}))$$

where P ranges over first order properties. Putting these two entries together, and using the rules of the λ calculus, we get:

$$(22) \quad \lambda \Phi \lambda \pi \Phi(\pi * \text{AG})(\lambda v \exists x (book(x, \pi * \text{BOOK}) \wedge \lambda y_1 \lambda \pi_1 (\text{enjoy}(v, y_1, \pi_1 * [\text{ARG}_2^{\text{enjoy}} : \text{EVT} - \epsilon(\text{HD}(\Phi), \text{BOOK})])))(x)(\pi * [\text{ARG}_2^{\text{enjoy}} : \text{EVT} - \epsilon(\text{HD}(\Phi), \text{BOOK})])))$$

The type presuppositions here cannot be satisfied as they stand. But *enjoy* licenses a natural transformation from objects to eventualities involving them. In the construction of logical form this introduces a *polymorphic type functor* with a polymorphic type that will serve to justify the basic type presupposition. The polymorphic type functor will apply to the λ abstract in the consequent given by the verb, $\lambda y_1 \lambda \pi_1 (\text{enjoy}(v, y_1, \pi_1))$. For type presuppositions, this is a general procedure for presupposition justification. The functor introduces a predicate related to the polymorphic type. When the polymorphic type is underspecified and of the form $\epsilon(\alpha, \beta)$, the corresponding predicate is $\phi_{\epsilon(\alpha, \beta)}(e, x, y)$. The functor instantiated on this example looks like this:

$$(23) \quad \lambda P \lambda u \lambda \pi'' (\exists z : \epsilon(\text{EVT}, \text{BOOK}) \exists z_1 : \text{AG}(P(\pi'')(z) \wedge \phi_{\epsilon(\text{AG}, \text{BOOK})}(z, z_1, u, \pi''))))$$

Applying the functor on the designated λ term, we get:

$$(24) \quad \exists x \exists z \lambda \pi (\text{enjoy}(j, z, \pi) \wedge book(x) \wedge \phi_{\epsilon(\text{AG}, \text{BOOK})}(z, j, x, \pi))$$

The functor in (23) suffices to handle event coercion with verbs whose type presuppositions are sensitive to both the type of the subject and object. There are many others involved in coercion. To handle the specification of the underspecified formula to the preferred reading, TCL relies on contextual rules and on defeasible generalizations about specifications. For example, in the absence of other information, we might specify $\phi_{\epsilon(\text{AG}, \text{BOOK})}(z, j, x, \pi)$ to *read*(z, j, x, π).

Subtyping in TCL remains subsumptive: if $\alpha \sqsubseteq \beta$ then anything of type α is something of type β , regardless of the context. As we want to leave the type of the variable associated with *book* unchanged, we must introduce a fresh term to refer to the eventuality that is introduced during the process of type justification. To give this a more mnemonic name, we call this process *spell out*. This view of subtyping contrasts with the approach we consider next.

4. Coercions in Modern Type Theories with Coercive Subtyping

Formal Semantics in MTTs. Modern Type Theories (MTTs)¹⁰ can be used as foundational languages for formal semantics. For instance, Ranta [31] has studied it in Martin-Löf’s type theory. This is in the tradition of the Montague grammar [23], although MTTs provide arguably better alternatives because of their rich type structures that are lacking in the Montague semantics. There is, however, an important difference: in an MTT common nouns (CNs) are interpreted as *types* rather than predicates.¹¹ For instance, the CNs like *table* and *man* are interpreted as types $\llbracket table \rrbracket$ and $\llbracket man \rrbracket$, rather than as predicates of type $E \rightarrow t$. Adjectives are then interpreted as predicates whose domains are not the type E of all entities, but the types over which the adjectives are meaningful. For instance, ‘handsome’ will be interpreted as a predicate of type $\llbracket man \rrbracket \rightarrow Prop$, where $Prop$ is the type of all logical propositions.

Interpreting CNs as types allows us to solve some problems in linguistic semantics. For instance, some linguistic features like copredication (see, for example, [3]) have been found difficult to deal with semantically in the Montagovian setting, but can be dealt with straightforwardly and satisfactorily in MTTs [14]: thanks to the fact that, in a formal semantics in MTTs, CNs are interpreted as types rather than predicates. However, there is a potential obstacle for MTTs to be employed for formal semantics: some notion of *subtyping* becomes essential since subtypes are crucially needed even for basic semantic constructions. Recently, it has been proposed that coercive subtyping provides us with such a subtyping framework for formal semantics [14, 17].¹² An MTT with coercive subtyping is a powerful language with rich type structures which provide a variety of useful mechanisms for formal semantics [15, 17].

Linguistic Coercions in MTTs with Coercive Subtyping. Coercive subtyping also provides us a framework to interpret various linguistic coercions in a formal semantics based on MTTs. The basic coercive subtyping mechanism¹³ 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, as exemplified in §2, the type-shifting law in [26] 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 CN (the universe that consists of the types that interpret CNs): $A <_{c[A]} (A \rightarrow Prop) \rightarrow Prop$, where $c[A](x) = \lambda P : A \rightarrow Prop. P(x)$. It can be verified that these coercions commute with the other subtyping relations and

¹⁰Examples of modern type theories include Martin-Löf’s type theory (MLTT) [20, 24], the Unifying Theory of dependent Types (UTT) [12] and the type theory implemented in the Coq proof assistant (pCIC) [9].

¹¹Further discussions on whether CNs should be interpreted as types can be found in [16], a recent article of the second author, where he argues for this based on Geach’s observation that CNs have their own criteria of identity [11], and the consequential necessity of adopting the principle of proof irrelevance in a formal semantics based on MTTs.

¹²It should be noted that coercive subtyping [13] was studied in 1990s for applications of MTTs to, for example, formalisation of mathematics and verification of programs, as supported by proof assistants such as Coq [9, 32], Lego [18, 5], Matita [21] and Plastic [7]. It is also worth noting that the traditional notion of subtyping, *subsumptive subtyping*, is not adequate for MTTs, while coercive subtyping is (the coercive subtyping extension is conservative, which implies among other things that the consistency of the original type theory is preserved) [19].

¹³Coercive subtyping as formulated in a logical framework [13, 19] also supports type-shifting in predicates.

hence satisfy the required coherence condition in coercive subtyping.

As another (formally simpler) example, let us reconsider (3), repeated here as (25):

(25) Julie enjoyed a book.

In an MTT, the formal interpretation of (25) is (26):

(26) $\exists x : \llbracket book \rrbracket . \llbracket enjoy \rrbracket(j, x)$

where

(27) $\llbracket enjoy \rrbracket : Human \rightarrow Event \rightarrow Prop.$

However, the domain type of $\llbracket enjoy \rrbracket(j)$ is *Event*, which is different from *Book*! Then, how can $\llbracket enjoy \rrbracket(j, x)$ in (26) be well-typed? The answer is that, in the framework of coercive subtyping and, in particular, under the assumption of the following coercion:

(28) $Book <_{reading} Event$

$\llbracket enjoy \rrbracket(j, x)$ is coerced into (and, formally, equal to) $\llbracket enjoy \rrbracket(j, reading(x))$ and hence well-typed. Informally, the sentence (25) is coerced into (29):

(29) Julie enjoyed reading a book.

Note that, in the above, we have considered only one possible coercion (28): from ‘enjoy a book’ to ‘enjoy reading a book’. As we noted in the previous section, however, there are in fact context-dependent ‘multiple coercions’: e.g., (25) could have meant ‘Julie enjoyed writing a book’ (cf., (3)); there could also be several reading events of that book. Coercive subtyping requires contextual uniqueness of coercions¹⁴, we must restrict the scope/context using *local coercions* [15].

Local Coercions. In many situations, it is necessary to limit the scope of a coercion. (25) furnishes an example: with the formal coercion (28), (26) is the correct interpretation of (25). However, there may be several possible coercions and hence (25) may have several meanings: which one to use can only be decided contextually. But note that coherence in coercive subtyping (contextual uniqueness of coercions) is necessary for formal semantics to deal with ambiguity. In such situations, we use local coercions to limit the scope of applicability of coercions. For instance, if (25) is used to mean (29) or ‘Julie enjoyed writing a book’, we exploit the following two coercions for (25):

(30) **coercion** $Book <_{reading} Event$ **in** (26)

(31) **coercion** $Book <_{writing} Event$ **in** (26)

¹⁴This refers to the notion of *coherence*, the requirement that any two coercions between the same two types (in the same context) be the same. See [13, 19] for its formal definition.

Note that such interpretations involve different local coercions and can be used in the same context. There is no ambiguity or confusion as to which coercion is to be employed, but we must make clear the scope of each one of the coercions, over what terms they are operative.

Local coercions have a dual notion – coercion contexts, which are contexts (in type theory) which may contain coercion entries of the form $A <_c B$ as well as entries of the usual form $x : A$. Coercion contexts occur left to the \vdash -sign. One can move a coercion entry in a coercion context to the right-hand side of the \vdash -sign to form a local coercion, while the inversion of this moves the coercion in a local coercion to the left. These constructs are governed by the relevant inference rules, some of which are discussed in, for example, [15].

Dependent Types in Coercion Semantics. Sometimes, a simple scoping restriction is not enough. For example, consider the following sentence (related examples are to be considered in (36)):

- (32) Jill just started *War and Peace*, which Tolstoy finished after many years of hard work.
But that won't last because she never gets through long novels.

It is not difficult to see that in (32) the scopes of the reading and writing coercions overlap intertwiningly, and so restrictions on the scopes of coercions will not be sufficient here to ensure uniqueness to eliminate ambiguity.

In many such cases, dependent typing proves to be useful. Indeed, this is the first time in the literature, as far as we know, that dependent types have been shown to be useful directly in the formal semantics of linguistic coercions.

For example, for the above sentences (32), instead of *Event*, we may consider the family of types

$$Evt : Human \rightarrow Type;$$

intuitively, for any $h : Human$, the dependent type $Evt(h)$ is the type of events conducted by h . Now, we can assume that the verbs *start* etc have the following types:

$$\begin{aligned} start, finish, last & : \Pi h : Human. (Evt(h) \rightarrow Prop) \\ read, write & : \Pi h : Human. (Book \rightarrow Evt(h)) \end{aligned}$$

Furthermore, we can consider the following coercions,¹⁵ for any $h : Human$,

$$Book <_{c(h)} Evt(h),$$

where the coercion $c(h)$ is the function from *Book* to $Evt(h)$ defined as follows: for any $b : Book$,

$$c(h, b) = \begin{cases} write(h, b) & \text{if } h \text{ wrote } b, \\ read(h, b) & \text{otherwise.} \end{cases}$$

¹⁵This is a *parameterised coercion*, parameterised by $h : Human$.

where we have simplified the second case by assuming that one would read a book if he/she has not written it. (One may think of other actions to consider more subcases here.) Having the above, we can now interpret (32) as follows (in a simplified form):

$$(33) \quad \begin{aligned} & start(j, wp) \\ & \& finish(t, wp) \\ & \& \neg last(j, wp) \\ & \& \forall lb : (\Sigma b:Book.long(b)). finish(j, \pi_1(lb)) \end{aligned}$$

where the Σ -type $\Sigma b:Book.long(b)$ is the type that interprets the CN ‘long book’ (a long book is a pair of a book together with a proof that it is long) and π_1 is the first projection operator that takes a long book and returns the book itself. In the coercive subtyping framework, (33) is coerced into (and equal to) the following:

$$(34) \quad \begin{aligned} & start(j, c(j, wp)) \\ & \& finish(t, c(t, wp)) \\ & \& \neg last(j, c(j, wp)) \\ & \& \forall lb : (\Sigma b:Book.long(b)). finish(j, c(j, \pi_1(lb))) \end{aligned}$$

which is (equal to)

$$(35) \quad \begin{aligned} & start(j, read(j, wp)) \\ & \& finish(t, write(t, wp)) \\ & \& \neg last(j, read(j, wp)) \\ & \& \forall lb : (\Sigma b:Book.long(b)). finish(j, c(j, \pi_1(lb))) \end{aligned}$$

Note that, in the last conjunct, the coercion c is still present – $c(j, \pi_1(lb))$ cannot be reduced furthermore because lb is a variable.

Remark The above example shows that dependent types provide powerful tools that can deal with rather sophisticated examples. It would be interesting to see exactly what powers they provide and, probably more interestingly, what they cannot deal with in formal semantics.

5. Are all corecions alike?

Both TCL and MTT approaches to coercions make available for subsequent discourse both ‘coerced content’ and the original meaning in a coercion. With the container/contents sort of examples, this strategy makes sense: both entities seem indeed to be available for subsequent anaphoric reference. It also makes sense for examples like (32). But this strategy doesn’t seem equally felicitous for all coercions.

- (36)
- a. Jill started writing a book. The writing will last for years.
 - b. Jill started a book. ?The writing /?That will last for years.
 - c. *Jill started *War and Peace*, which will last for weeks.
 - d. Jill started to read the article thoroughly.
 - e. *Jill started the article thorough/ thoroughly.
 - f. John is scared of starting *War and Peace*, because it will last for (will take him) weeks.

Anaphoric reference to the coerced eventualities that spell out and MTT with dependent types introduce into logical form is not felicitous in (36b), in contrast to (36a). Further, the use of an appositive relative clause that is supposed to modify the event introduced in the coercion is not grammatical, and neither is adjectival event modification in (36e). Non-restrictive relative clauses are only licensed if the DP they modify denotes an entity of the sort the relative clause can apply to [27]. Examples like (36c,d) show that the coerced eventuality is neither available for coreference nor modification, indicating that it is not present in our semantic logical forms.

Surprisingly, anaphoric reference to the eventuality becomes better when certain discourse patterns are introduced, in particular causal relations like Explanation (36f) and Contrast relations (32), but not in cases where we are continuing a topic or in an Elaboration (36b,c). Thus, the predictions of spell out and MTT with dependent types are incorrect except when there is a particular discourse configuration between the clause in which the eventuality referring pronoun occurs and the clause containing the event coercion.

Some coercions behave even more bizarrely. In these examples inspired by [25], the same discourse sensitivity is observed, but the *original denotation* of the argument seems unavailable except in the aforementioned discourse patterns.

- (37)
- a. ??The omelette left without paying, although it was very yummy.
 - b. ??The omelette, which has fresh mushrooms in it, left without paying.
 - c. ? The delicious omelette has left without paying.
 - d. The omelette has ordered. He wants it with mushrooms.
 - e. The omelette is getting restless. And it's burned, because it cooked too long.
 - f. The omelette left without paying because he found it disgusting.

The impossibility of using non restrictive relative clauses (37c) or even certain non restrictive adjectival modifications that require the DP to have its standard meaning (37d) are very surprising. The aspectual verbs and *enjoy* allow non restrictive modification of the *standard meaning* of their arguments; e.g., *Julie enjoyed the novel, which is a thrilling murder mystery*. According to the evidence, *the omelette* (37a-c) simply doesn't refer to the food stuff at all. Nevertheless, with suitable discourse configurations involving particular discourse relations, as in (37e-h), anaphoric reference to the foodstuff, the original meaning of the *omelette* is felicitous.

The dependence of anaphoric availability on discourse structure echoes an observation in [10] concerning the availability of individual discourse referents for anaphoric coreference in bare nominal expressions in Catalan and Spanish.¹⁶ They observe that a bare singular noun complement, in contrast to a normal indefinite DP, does not always introduce an individual level discourse referent that is available for subsequent anaphoric reference:

- (38) a. Avui porta faldilla. #La hi vam regalar lany passat.
Today she is wearing a skirt. We gave it to her as a present last year.
- b. Avui porta una faldilla. La hi vam regalar lany passat.
Today she is wearing a skirt. We gave it to her as a present last year.

Non restrictive relative clauses modifying an individual type entity are also not possible with bare singular NPs, though they are of course good with full DPs:

- (39) *Per fi hem trobat pis, que comencarem a reformar molt aviat.
At last we have found an apartment, which we will begin to renovate very soon.

Thus, bare singular NPs in Spanish, Catalan and Norwegian have a similar referential opacity to many coercions.

However, [10] also note that discourse anaphora using the pronouns *la* and *el*, which require their antecedents to denote individual entities, with bare singular NPs is possible in some discourse contexts. Again we give their examples:

- (40) a. Per la festa es va posar faldilla. Se l'havia comprat la tarda anterior.
She put on a skirt for the party. She had bought it the day before in the afternoon.
- b. Ja tinc pis. El vaig comprar ahir.
I already have an apartment. I bought it yesterday.

Once again we note one of the discourse configurations, Explanation, that licenses anaphoric reference to coerced eventualities. And apparently Contrast also supports the felicitous use of individual level pronouns with bare singular NPs.¹⁷

6. Accounting for the Data

One may consider the similarity between coercion insertion in coercive subtyping and TCL's spell out mechanisms. They both deliver discourse entities that should be available anaphorically. The data shows that this is not always true. [10] suggest that some sort of discourse accommodation might be at work in the cases where anaphoric reference to individual level discourse entities is licensed by bare NPs, while standard semantics makes them denote properties. The data involving coercion, however, suggests a different explanation. We need the arguments of the coercing predicates to be of the appropriate type, and so we cannot simply let ordinary compositional semantics

¹⁶They also note that similar observations have been made for bare singular NPs in Norwegian[6].

¹⁷Louise McNally, pc.

“do its job”, because ordinary semantics is not equipped to build proper logical forms for coercing predicates. We need to use some coercion mechanism to allow composition to succeed but not to always produce discourse entities associated with coerced meaning.

Our solution is to identify two coercion mechanisms. The first is a *local* coercion that adjusts types only within an argument of the predicate licensing the coercion as in the MTT-approach (cf., §4). As with simple coercive sub typing, this coercion has a very local scope. We can implement both local coercion and spell out in TCL by introducing a functor which takes a term t , an input type a and an output type b of the following form:

$$(41) \quad \lambda P \lambda x C_{t,a,b}(P(x))$$

Applying this functor instead of the one associated with Spell Out, the logical form for (3), ignoring the type presupposition parameters π for readability, becomes:

$$(42) \quad \exists x(\text{book}(x) \wedge C_{wp,\text{BOOK,EVT}} \text{enjoy}(j, x))$$

The operator $C_{wp,\text{BOOK,EVT}}$ requires a semantics. Assuming a background dynamic semantics like DPL or compositional DRT, we also assume a function c associated with our coercion that maps objects of type a into objects of type b . Given this, the semantics of our operator is relatively straightforward.

$$(43) \quad \mathfrak{A}, w, f \parallel C_{t,a,b} \phi(t_1, \dots, t_n, t) \parallel \mathfrak{A}, w, f \text{ iff} \\ \exists g \text{Dom}(g) = \text{Dom}(f) \cup \{c(t)\} \wedge \mathfrak{A}, w, f \parallel \phi(t_1, \dots, t_n, c(t)) \parallel \mathfrak{A}, w, g$$

This allows type presuppositions in TCL to be justified, because the argument for the predicate is mapped into something of the right type. And so predication succeeds. However, this entity is *not* represented by a distinct term in logical form and so, given standard assumptions of dynamic semantics about the treatment of anaphora, is not available as an antecedent for anaphora in the discourse to come.

To handle coercions of the omelette variety, we have to change coercion operators. Here we suppose an operator $H_{t,a,b}$ (for *hide*) that provides a map for t from type a to type b . However, here only the result of the map is visible in logical form. For a sentence like (37a), we would have (ignoring the presuppositional nature of the definite, which poses some interesting problems in itself):¹⁸

$$(45) \quad \exists x \exists z H_{\text{the-omelette,FOOD, PERSON}}[\text{omelette}(x) \wedge \text{ate}(x, z)] \wedge z = ?$$

¹⁸The semantics of $H_{t,a,b}$ is as follows:

$$(44) \quad \mathfrak{A}, w, f \parallel H_{t,a,b} \phi(t_1, \dots, t_n, t) \parallel \mathfrak{A}, w, f \text{ iff} \\ \exists g \text{Dom}(g) = \text{Dom}(f) \cup \{c^{-1}(t)\} \wedge \mathfrak{A}, w, f \parallel \phi(t_1, \dots, t_n, c^{-1}(t)) \parallel \mathfrak{A}, w, g$$

Our treatment entails that the map from a to b must be bijective. Our theory predicts that there are no omelette type coercions where there is not a one to one correspondence between the objects of the coerced type b and the objects of the original type a .

We suppose local coercions that come in several flavors and that are tied to particular types of coercions like the eventuality based coercions or the omelette type coercions. The second mechanism we suppose is that of spell out. For coercions like container/contents coercion, spell out seems the right one. For the others considered in this paper, however, local coercions are appropriate as the basic mechanism. When certain, “anti-local” discourse configurations obtain, spell out takes these local coercions and replaces them with the more elaborate functors of the sort provided in (23), which introduces a new discourse entity of the new type into the discourse context and that is available for subsequent anaphoric reference. Schematically, the transformation is this:

$$(46) \quad C_{t,a,b}F(t_1, \dots t_n, t) \rightsquigarrow \exists x : b(Rxt \wedge \phi(t_1, \dots t_n, x)) \text{ (where } t : a).$$

We replace our formula prefixed with a local functor with a first order formula, R being specified by the polymorphic type associated with C .¹⁹

Much the same story can be told for the data concerning bare singular NPs. Assuming with Espinal and McNally that bare singular NPs denote properties, we imagine that certain discourse configurations can license the application of a functor to the singular NP producing an entity realizing that property.²⁰ Thus, we would explain the observations about anaphoric availability of individual discourse entities with bare NPs as a species of spell-out, though the spell out in these cases would work off the basic semantics assigned to bare NPs, not off a coercion operator.

What controls spell out? What is the anti-local feature that controls or licenses spell-out? The data show that this licensing depends on particular discourse configurations, not sentential distance. It also depends on particular coercions that themselves depend on the predicate at issue. We suppose that the basic coercion form with operators like C and H initially introduced in logical form by the type adjustment and the type requirements of particular predicates are transformed via Spell Out into the more familiar material from TCL and MTT when an anti-local discourse configuration occurs. For the event based coercions, it is crucial that the discourse unit with the event anaphor relates to the unit with the coercion via a relation that enforces that the eventuality introduced in the coercion is distinct from the eventuality described by the clause with the anaphor. This is guaranteed for example by a relation like Explanation or Result, whose semantics involves a causal relation; if there is a causal relation between a and b , then a and b must be distinct because there are no self-causing facts or events. Contrasts also involve a shift from one fact to another that perform have different properties. Relations like Elaboration, where $\text{Elaboration}(a, b)$ holds only if b expands on or gives more details about the fact or event described in a do not guarantee distinctness, and a relation like Continuation has a semantics according to which a and b are both descriptions of the same topical event or fact.

All of the examples in which spell out makes the correct predictions about anaphoric availability

¹⁹For our functors of type H , we suppose a slightly different form of spell out., which we do not elaborate on here for lack of space.

²⁰For a discussion of functors that produce realizations of entities of abstract type, see [3].

with event coercions involve one of these anti-local discourse relations. The examples in which spell out makes the wrong predictions are those that feature relations like Elaboration, Reformulation or Continuation or feature the anaphor and its antecedent within one elementary discourse unit. Consider (36b) or (36b) the second sentence, or appositive relative clause, attempts to elaborate on the coerced event. The discourse structure between the two clauses does not have the requisite element of non locality. On the other hand, (36f,g) involve relations of Flashback (a kind of reverse narrative sequence where the second related fact precedes and is spatiotemporally distinct from the first), Contrast and Explanation.

The data in (37) forces us to refine our anti-locality hypothesis. (37b) also features a relation of Elaboration between the clause with the coercion and the relative clause with the anaphor. We correctly predict that example to be infelicitous, We also predict examples (37d-f) to be good because they feature anti-local discourse structures. But (37d), which has an Elaboration relation between the two clauses, is good. However, this is not an Elaboration of the *original meaning* of *the omelette*; it's an Elaboration of what the omelette *qua* individual has ordered. And this makes all the difference. This also explains why (37c) is no good. While it has contrast built into it—it is what many researchers on discourse structure would call a Concession, it is also a Background (in SDRT concessions are defined in terms of Contrast and Background), and Background requires a topic, which in this case is also the original denotation of *the omelette*. What is common to all of the bad or local discourse structures is that the element hidden by the local coercion operators is in the discourse topic dominating the discourse constituent in which the anaphor to the hidden element occurs. The bad cases of Elaboration all have the hidden discourse entities in topic position, as does (37c). Our hypothesis about when spell out is licensed is: spell out for coercions with the non robust coercions occurs only in those discourse structures in which the element hidden by the local coercion operators is not in the topic position (for details on topics in discourse structure, see for instance [1, 4]).

Why should such a constraint on spell out exist? Why should it only exist for some coercions? The container/content coercions are happy in English to provide discourse entities corresponding to the original meaning of the coercing predicate's argument and discourse entities that are inferred from the type coercion mechanism in the local discourse configurations. Because of its sensitivity to the type of coercion at issue, we do not believe that the anti-locality constraint is a general feature of discourse structure. It is rather a feature of particular coercions and of particular languages. More research on the varieties of coercions is needed, especially across different languages.

7. Conclusions

Coercions are not all alike. Container/contents coercions make standard and coerced content available for future discourse exploitation. But eventuality based coercions and others don't. The typology of coercions with respect to their behavior on discourse continuations seems rich and in need of investigation, especially cross-linguistically. However, the frameworks that we have proposed are robust enough and precise enough to permit an investigation of this typology. It may be

interesting to study the relationship between TCL and MTTs. In future work, we hope to explore this further as well as the rich and delightful world of coercions cross-linguistically.

References

- [1] Nicholas Asher. *Reference to Abstract Objects in Discourse*. Number 50 in Studies in Linguistics and Philosophy. Kluwer, Dordrecht, 1993.
- [2] Nicholas Asher. A Type Driven Theory of Predication with Complex Types. *Fundamenta Informaticae*, 84(2):151–183., 2007.
- [3] Nicholas Asher. *Lexical Meaning in Context: A Web of Words*. Cambridge University Press, 2011.
- [4] Nicholas Asher and Alex Lascarides. *Logics of Conversation*. Cambridge University Press, 2003.
- [5] Anthony Bailey. *The Machine-checked Literate Formalisation of Algebra in Type Theory*. PhD thesis, University of Manchester, 1999.
- [6] Kaja Borthen. *Norwegian bare singulars*. PhD thesis, Norwegian University of Science and Technology, 2003.
- [7] Paul Callaghan and Zhaohui Luo. An implementation of typed LF with coercive subtyping and universes. *J. of Automated Reasoning*, 27(1):3–27, 2001.
- [8] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940.
- [9] The Coq Development Team. *The Coq Proof Assistant Reference Manual (Version 8.3)*, INRIA, 2010.
- [10] Teresa Espinal and Louise McNally. Bare nominals and incorporating verbs in spanish and catalan. *Journal of Linguistics*, 47(1):87–128, 2011.
- [11] P. Geach. *Reference and Generality*. Cornell University Press, 1962.
- [12] Zhaohui Luo. *Computation and Reasoning: A Type Theory for Computer Science*. OUP, 1994.
- [13] Zhaohui Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1):105–130, 1999.
- [14] Zhaohui Luo. Type-theoretical semantics with coercive subtyping. *SALT20, Vancouver*, 2010.
- [15] Zhaohui Luo. Contextual analysis of word meanings in type-theoretical semantics. *LACL'11, LNAI 6736*, 2011.

- [16] Zhaohui Luo. Common nouns as types. In D. Bechet and A. Dikovsky, editors, *Logical Aspects of Computational Linguistics (LACL'2012)*. LNCS 7351, 2012.
- [17] Zhaohui Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 2012. (to appear).
- [18] Zhaohui Luo and Robert Pollack. LEGO Proof Development System: User's Manual. LFCS Report ECS-LFCS-92-211., Department of Computer Science, University of Edinburgh, 1992.
- [19] Zhaohui Luo, Sergei Soloviev, and Tao Xue. Coercive subtyping: theory and implementation. *Information and Computation*, 223 (2013), 2012.
- [20] Per Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.
- [21] The Matita proof assistant. <http://matita.cs.unibo.it/>, 2008.
- [22] John Mitchell. Coercion and type inference. In *Proc. of Tenth Annual Symposium on Principles of Programming Languages (POPL)*, 1983.
- [23] Richard Montague. *Formal Philosophy*. Yale University Press, 1974. (Collection edited by R. Thomason).
- [24] Bengt Nordström, Kent Petersson, and Jan Smith. *Programming in Martin-Löf's Type Theory*. Oxford University Press, 1990.
- [25] Geoffrey Nunberg. Transfers of meaning. *Journal of Semantics*, 12:109–132, 1995.
- [26] Barbara Partee and Mats Rooth. Generalised conjunction and type ambiguity. In Bauerle, Schwarze, and von Stechow, editors, *Meaning, Use, and Interpretation of Language*, 1983.
- [27] Christopher Potts. *The logic of conversational implicature*. Oxford University Press, 2005.
- [28] Stephen Pulman. Aspectual shift as type coercion. *Transactions of the Philological Society*, 52(2), 1997.
- [29] James Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.
- [30] James Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.
- [31] Aarne Ranta. *Type-Theoretical Grammar*. Oxford University Press, 1994.
- [32] Amokrane Saïbi. Typing algorithm in type theory with inheritance. *Proceedings of Principles of Programming Languages 1997*, 1997.