Copredication in homotopy type theory

Hamidreza Bahramian, Narges Nematollahi, Amr Sabry

Copredication in homotopy type theory

HAMIDREZA BAHRAMIAN, Indiana University Bloomington
NARGES NEMATOLLAHI, Indiana University Bloomington
AMR SABRY, Indiana University Bloomington

This paper applies homotopy type theory to formal semantics of natural languages and proposes a new model for the linguistic phenomenon of copredication. Copredication refers to sentences where two predicates which assume different requirements for their arguments are asserted for one single entity, e.g., "the lunch was delicious but took forever". This paper is particularly concerned with copredication sentences with quantification, i.e., cases where the two predicates impose distinct criteria of quantification and individuation, e.g., "Fred picked up and mastered three books." In our solution developed in homotopy type theory and using the rule of existential closure following Heim analysis of indefinites, common nouns are modeled as identifications of their aspects using HoTT identity types, e.g., the common noun *book* is modeled as identifications of its physical and informational aspects. The previous treatments of copredication in systems of semantics which are based on simple type theory and dependent type theories make the correct predictions but at the expense of ad hoc extensions (e.g., partial functions, dot types and coercive subtyping). The model proposed here, also predicts the correct results but using a conceptually simpler foundation and no ad hoc extensions. The proofs in the proposal have been formalized using Agda.

Additional Key Words and Phrases: homotopy type theory, formal semantics of natural languages, computational semantics, copredication, programming and proving in Agda

1 INTRODUCTION

In formal linguistics the organization of grammar is viewed as involving a number of (relatively) independent subsystems, including phonology, which deals with the sounds of the language, morphology, which studies the smallest units of meaning in the language, syntax, which focuses on how the units combine to make a grammatical sentence, and semantics, which is the focus of this paper and studies the meaning of a grammatical sentence based on the meaning of its constituents. In theories of formal semantics, the assumption is that the meaning of the sentence is related to the meanings of its parts in a systematic way. This assumption which is usually attributed to the 19th century German logician, Gottlob Frege, is called the *Principle of Compositionality*, and can be formulated as follows:

> Principle of Compositionality: The meaning of a compound expression is a function of the meanings of its parts and of the way they are syntactically combined [Partee 1984].

Considering the Principle of Compositionality, the task of theories of formal semantics is twofold: 1) defining the lexicon by providing the semantic denotations for the constituents, and 2) setting some rules which combine the constituents as defined in the lexicon and return the truth conditions of the sentences. In the framework of formal semantics, the semantic denotation of a linguistic expression is represented by $[\![\cdot]\!]$ which is called the *Interpretation Function* and which maps linguistic expressions to their semantic denotations. For example, if α is a linguistic expression, $[\![\alpha]\!]$ is the semantic denotation of α.

As a brief review of the systems of formal semantics that have been developed so far, the pioneering work of Montague[1] [Montague 1975] as well as all the later works in the Montague tradition use Church's simple type theory [Church 1940]. More recent work (e.g., [Luo 2012b; Ranta 1994]) employ dependent type theories like Martin-Löf's type theory [Nordström et al. 1990] and the Unifying Theory of dependent Types [Luo 1994].

The copredication phenomenon has been recently the topic of many discussions in the field of formal semantics (e.g., [Asher 2008, 2011; Bekki and Asher 2012; Chatzikyriakidis and Luo 2015; Gotham 2014; Retoré 2014]). Copredication sentences are those where two predicates with different requirements on their arguments are asserted for one single entity. For example, sentence (1c) below represents a copredication sentence:

(1) a. The lunch was delicious.
    b. The lunch took forever.
    c. The lunch was delicious and took forever.

Delicious is a predicate which is normally used of food but not events, while *took forever* is a predicate which normally holds of events and not of food. As Cooper [Cooper 2011] points out, if we were only dealing with sentences like (1a) or (1b), we could present them as instances of polysemy by saying that *lunch* is ambiguous between a food interpretation and an event interpretation, or in terms of types, we could say that *lunch* in some cases is of type *Food*, and in some others of type *Event*. However, cases like (1c) where one occurrence of the word simultaneously has both interpretations gives rise to the question as what type should we assume for *lunch*.

For the purpose of this paper, we distinguish between three kind of lexical ambiguity: homophony, underspecified[2] and metaphor: homophony refers to cases where two semantically different words happen to have the same phonological form (e.g., *bank* as office or as land), underspecified refers to the phenomenon that one and the same word is being considered through its different aspects (e.g., *book* as its physical aspect or as its informational aspect), and metaphor refers to cases where a word is used in place of another word by virtue of some semantic relation between the two (e.g., *newspaper* as the institution publishing newspapers). What we are considering in this work is the case of underspecified as it seems it is where felicitous copredications are possible.

In what follows, first we will look at the treatment of the problem of copredication in Montague's system (Sec. 2), in a modified [Heim and Kratzer 1998] version of Montague system (Sec 3), and in formal semantics based on modern type theories (Sec 4). After pointing out the shortcomings of each solution, we will then propose a model based on homotopical interpretation of identity types in pure intensional type theory, and show that it can adequately address the complexity of copredication sentences.

2 COPREDICATION IN MONTAGUE SEMANTICS

In Montague's system there are only two basic types: the type e which represents entities, and the type t which represents truth values. D_e refers to the set of all entities, and D_t is the set of all truth values, consisting of 0 for false sentences and 1 for true sentences. All linguistic items are taken as functions of these two basic types. Following the established notation, the type of functions from D_e to D_t is $\langle e, t\rangle$. Thus, in the case of sentence (1), *delicious* and *took forever* are taken as predicates of type $\langle e, t\rangle$, and the connective *and* is defined as a function of type $\langle\langle e, t\rangle, \langle\langle e, t\rangle, \langle e, t\rangle\rangle\rangle$.

[1]The Montagovian setting uses a logic for meaning assembly(simply typed lambda calculus) and a logic for semantic representation(higher-order predicate logic)[Moot and Retoré 2012].

[2]We borrow this term from [Zwicky and Sadock 1975], where one of the tests to distinguish between underspecified and polysemous (which constitutes homophony and metaphor) is the felicity of relevant copredications.

For the sake of simplicity and since it has no direct bearing on our discussion, we do not include tense in our analysis, and treat *took forever* as an atomic predicate rather than the past tense form of *take* plus the temporal adverb *forever*. The copula (*was*), on the other hand, is taken as semantically vacuous, so *delicious* and *was delicious* have the same semantic values. Therefore, the lexicon of sentence (1) is defined as follows:

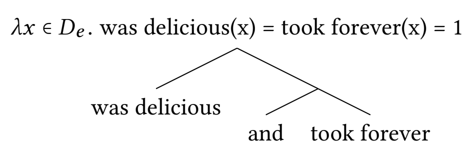
$$\begin{aligned}
[\![\text{was delicious}]\!] &= \lambda x \in D_e.\ x \text{ was delicious} \\
[\![\text{took forever}]\!] &= \lambda x \in D_e.\ x \text{ took forever} \\
[\![\text{and}]\!] &= [\lambda f \in D_{\langle e,t\rangle}.[\lambda g \in D_{\langle e,t\rangle}.[\lambda x \in D_e.\ f(x) = g(x) = 1]]] \\
[\![\text{lunch}]\!] &= \lambda x \in D_e.\ x \text{ is a lunch} \\
[\![\text{the}]\!] &= \lambda f \in D_{\langle e,t\rangle}.\ !y\ [(f(y) = 1) \wedge (\exists! x \in D_e\ [f(x) = 1])], \\
&\quad \text{where } \exists! x\ [\phi] \text{ abbreviates “there is exactly one } x \text{ such that } \phi\text{”} \\
&\quad \text{and } !y\ [\phi] \text{ returns “that unique } y \text{ such that } \phi\text{”.}
\end{aligned}$$

Regarding the rules which combine the constituents of the lexicon, the only rule that we need for sentence (1) is *Functional Application* which is formulated as follows:

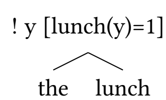
(2) Functional Application (FA) [Heim and Kratzer 1998]: if α is a branching node with β and γ as its daughters, then α is in the domain of $[\![\cdot]\!]$ if both β and γ are, and if $[\![\gamma]\!]$ is in the domain of $[\![\beta]\!]$. In this case, $[\![\alpha]\!]=[\![\beta]\!]([\![\gamma]\!])$.

Applying Functional Application[3] to the lexicon defined above, we will have the following denotations for the upper nodes: (a) calculates the semantic value of *was delicious and took forever* where *and* takes the two predicates *was delicious* and *took forever* as its two arguments and returns a function of type $\langle e, t\rangle$. (b) calculates the semantic value of *the lunch* where *the* takes *lunch* as its argument and returns the unique lunch the details of which are shared by participants of the conversation. Finally (c) calculates the semantic value of the whole sentence, returning 1 iff the lunch was delicious and took forever.

(3) a.

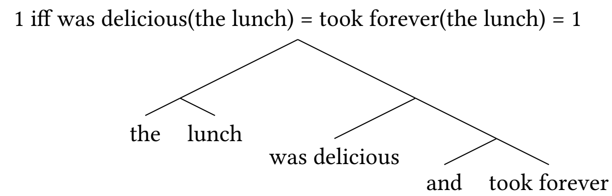


b.



[3] We can apply FA to a node if one of its children is an argument to the other. Based on the definition of FA, it does not matter whether the argument is the left child or the right one.

c.



The previous example suggests that the Montague semantics can handle the copredication sentence in (1c) without difficulty. However, a more precise analysis reveals that Montague semantics makes too many identifications. More precisely, all individuals are assigned type e, and consequently predicates always take arguments of type e; *the lunch* for example is assigned type e no matter if it is used in its food or event sense and *delicious* takes an argument of type e even though we know that *delicious* does not make sense with the event aspect of *lunch*. As a consequence, while Montague semantics avoids some possible problems in copredication sentences such as (1c), it is unable to provide the correct semantics in examples including quantification where the two predicates impose distinct criteria of individuation [Gotham 2014]:

(4) a. Fred picked up three books.
b. Fred mastered three books.
c. Fred picked up and mastered three books.

The predicate *picked up* deals with the physical aspect of books, whereas *mastered* is concerned with the informational aspect of books. We observe that the two predicates impose distinct criteria of individuation on their arguments: on one hand, sentence (4a) is true iff Fred picked up three books that are physically distinct, even if they are for example three copies of the same book, i.e., being informationally the same. On the other hand, sentence (4b) is true iff Fred mastered three books that are informationally distinct even if all three are contained in a trilogy, which is counted as one physical object. Combining the two predications, sentence (4c) is therefore true iff Fred picked up and mastered three books which are both physically and informationally distinct. In order to see if Montague semantics can handle sentences like (4), we first define the lexicon:

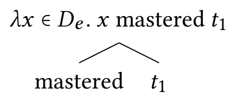
$$\begin{array}{rcl} [\![\text{Fred}]\!] & = & \textit{Fred} \\ [\![\text{picked up}]\!] & = & \lambda x \in D_e.\lambda y \in D_e.\ y \text{ picked up } x \\ [\![\text{mastered}]\!] & = & \lambda x \in D_e.\lambda y \in D_e.\ y \text{ mastered } x \\ [\![\text{book}]\!] & = & \lambda x \in D_e.\ x \text{ is a book} \\ [\![\text{three}]\!] & = & \lambda f \in D_{\langle e,t\rangle}.\lambda g \in D_{\langle e,t\rangle}.\exists x_1, x_2, x_3 \in D_e \text{ such that } x, y, z \text{ are distinct and} \\ & & [f(x_i) = g(x_i) = 1 \text{ for i= } 1, 2, 3] \end{array}$$

The syntactic tree for (4b), shown in (6d) exhibits the accepted practice of quantifier raising in the case of quantified objects [Heim and Kratzer 1998], where the quantified object is raised to a higher node, and leaves behind a trace (t_i) of type e. Then, in order to relate the trace with the raised object, we add a branch to the tree which has only a numerical index, co-indexed with the trace, right below the raised object. So in the case of our example, in (6d), first we raise the object *three books* to a higher node, and call its trace t_1. We note that t_1 is of type e, and composes with its neighbors just like any other element of type e. Then, in order to relate the raised object to its trace, we add a branch with the numerical index 1, co-indexed with the trace, right below the raised object. Now we calculate the semantic value of the tree step by step. (a) and (b) apply Functional Application and calculate, respectively, the semantic value of *mastered* t_1

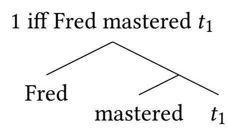
and *Fred mastered* t_1. But to calculate (c), we need a new compositional rule, called *Predicate Abstraction* (PA) shown in (5). This rule essentially says that in calculating the semantic value of a certain point of a tree, if we encounter a branch with a numerical index (like branch 1 in (6d)), which reflects that we have a raising operation, then we need to replace in the semantic value of that point of the tree the trace of the same index (i.e., t_1 in (6d)) with the variable x. By so doing, we make the semantic value into a function that can take the raised object as its argument and thereby, replace the trace with the raised object. So in the final step, shown in (d) the raised object *three books* is composed with the lower node, which is now a function expecting an argument of type e, through Functional Application.

(5) Predicate Abstraction (PA) [Heim and Kratzer 1998]: Let α be a branching node with β and γ as its daughters, where β dominates only a numerical index i. Then for any variable assignment g, $[\![\alpha]\!]^g = \lambda x \in D_e.\ [\![\gamma]\!]^{g[i \to x]}$.

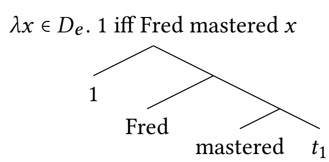
(6) a.



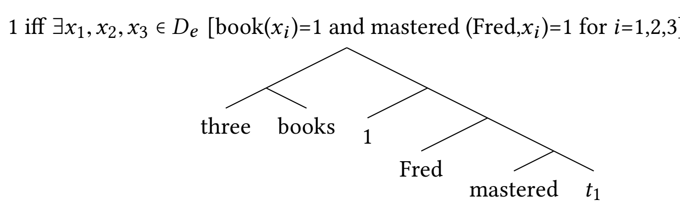
b.



c.



d.



As (6d) shows, in Montague semantics (4b) is true iff there exist three books x_1, x_2, x_3 which Fred mastered. There is no explicit restriction to establish how the three books are individuated, i.e., physically, informationally or both. Consequently, it is unclear how the system responds to scenarios where Fred mastered a trilogy contained in one single volume. The same unclear situation arises in response to sentence (4a), which requires three physically distinct books, and the copredicated sentence in sentence (4c) which needs three physically and informationally distinct books.

3 COPREDICATION IN MODIFIED MONTAGUE SEMANTICS

In the semantic framework of Heim and Kratzer [Heim and Kratzer 1998], which is based on Montague semantics, the requirements imposed by predicates on their arguments is modeled by taking the predicates as partial functions. So, *delicious* in (1) is no longer a total function from the set of individuals (of type e) to truth-values (of type t); rather, it is a partial function defined only for a subset of individuals, i.e. only for things with a food property so that *delicious* can be applied to them. Similarly, *took forever* is a partial function defined only for the subset of individuals with an event property. One of the main motivations for taking the predicates as partial functions in this framework is to be able to rule out sentences like *the chair laughed*, because the predicate *laugh* in this framework is undefined for inanimate entities. In the case of our examples, for the compositions in (1a) and (1b) to go through, *the lunch* in (1a) must be an entity belonging to the domain of *delicious* and in (1b) to the domain of *took forever*. Heim and Kratzer do not go into details about the possible operators between partial functions, but we can deduce that for the composition of (1c) to proceed, *the lunch* needs to be in the intersection of the two domains, so it needs to be both an entity of food property and that of event property, which is indeed the case.

In (4b) *mastered* is taken as a partial function defined for entities with an informational property and undefined otherwise. Therefore, (4b) is true iff there exist 1) three distinct entities (i.e. three distinct elements of type e) such that 2) each one is a book (i.e. the predicate ($\lambda x.x$ is book) returns true for each one of the entities) and 3) each one is informational (so that for each one the predicate *mastered* can be defined). These three conditions confirm that there are three distinct books, but do not guarantee that the three books are informationally distinct. The problem here, just like in Montague semantics, is that distinction is attributed to elements of type e, the only type for individuals that we have in simple type theory, and it is not clear how to define the criteria of distinction for elements of type e. While "three distinct informational entities" or "three distinct physical entities" can be clearly defined, when it comes to nouns with more than one aspect, like the common noun "book", the term "three distinct books" is ambiguous. What does it mean for three books to be different from each other? Does it mean that they are physically distinct or informationally distinct? This shows that taking predicates as partial functions cannot help with the problem of criteria for quantification; Heim and Kratzer's semantics has the same problem as Montague semantics in that it cannot produce the correct truth condition for a sentence which involves quantification, in simple and copredicated sentences.

To sum up our discussion so far, we saw that the main problem is to define the criteria of quantification for nouns that have more than one aspect. So in the case of the common noun "book" characterized by two aspects, physical and informational, we can either say that three books are distinct when they are distinct in just one of the aspects or when they are distinct in both aspects. In the former case, if we say that they are distinct when they are physically different, then the system wrongly returns true for "Fred mastered three books" even if the books are informationally similar. On the other hand, if we say that books are distinct when they are different in both physical and informational aspects, the system wrongly returns false for "Fred mastered three books" if they are contained in one single volume of a trilogy. We observe that what is needed is to develop a dynamic criterion of quantification which is defined based on the predicate.

4 COPREDICATION IN EXTENDED TYPE THEORIES

In theories of formal semantics which employ multiple-sorted type theories, sortal requirements that are imposed by predicates on their arguments are encoded in the type system [Luo 2012a; Ranta 1994]. Consequently, these systems are characterized by a richer inventory of types than in Montague semantics, since for example, instead of type e, we have various types *Physical*, *Animate*, *Inanimate*, etc., which represent respectively physical, animate and inanimate entities.

Furthermore, common nouns are also defined as types, and not functions of type $\langle e, t\rangle$ as in Montague semantics[4]. So for example, *book* denotes the type *Book*. Extending the inventory of types, however, brings about some type mismatch problems which require additional tools/extensions to type theory to resolve.

As a simple example, in a sentence like (7), *John* could be of type *Man*, but *shout* as a predicate would take an argument of type *Human*, and thus the semantic system encounters a type mismatch and wrongly predicts that the sentence is uninterpretable.

(7) John shouts.

In order to avoid such problems, Luo [Luo 1999] defines the notion of *coercive subtyping*, symbolized by $<_c$ as follows: for two types A and B, A $<_c$ B indicates that there is a unique implicit coercion from type A to type B in the sense that an object of type A can be used in any context requiring an object of type B. In sentence (7), *Man* is a subtype of type *Human*, and therefore, the coercive subtyping rule tells us that *John* of type *Man* can be composed with *shout* which requires an object of type *Human*.

A related extension to type theory is the notion of *dot types*, which is introduced by Pustejovsky [Pustejovsky 1991] in his treatment of copredication sentences, and is defined as follows: dot types are compositions of two types which nevertheless allow the two individual types to be recovered. Luo [Luo 2012b] then includes his notion of coercive subtyping into dot types and asserts the following statement about dot types:

> A.B is only well-formed if A and B do not share common components, and both projections, one from A.B to A and the other from A.B to B, are coercions in the coercive subtyping framework.

Sentences in (1), for example, involve a type defined as *Food.Event*, for which the following statements hold:

(8) a. *Food.Event* $<_c$ *Food*

b. *Food.Event* $<_c$ *Event*

According to the definition of coercive subtyping, if we include the type *Lunch*, we will have:

a. *Lunch* $<_c$ *Food.Event* $<_c$ *Food*

b. *Lunch* $<_c$ *Food.Event* $<_c$ *Event*

Then, by means of contravariance for function types, we derive the following relationships from rules in (8):

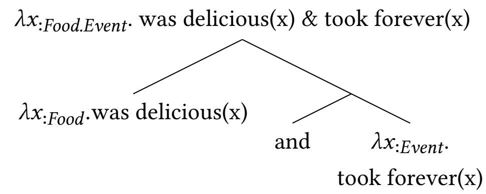
(9) a. (*Food* → *Prop*) $<_c$ (*Food.Event* → *Prop*) $<_c$ (*Lunch* → *Prop*)

b. (*Event* → *Prop*) $<_c$ (*Food.Event* → *Prop*) $<_c$ (*Lunch* → *Prop*)

Considering the subtyping relationships shown in (9), we can now analyze sentences (1a-1c): in (1a), *was delicious*, defined as $[\![\text{was delicious}]\!] = \lambda A_{:Food}.A$ was delicious, needs an object of type *Food*, but *the lunch* is an object of type *Lunch*. The subtyping relationship in (8a), however, says that an object of type *Lunch* can be used in any context which requires an object of type *Food*, and thus the composition can proceed. The same thing applies to (1b). As for the copredication sentence in (1c), *delicious* and *take forever* are identified as instances of "conjoinable types" [Chatzikyriakidis and Luo 2013], i.e., they can be coerced to the common type *Food.Event* → *Prop*. Therefore, as shown in (10), *was delicious and took forever* is a predicate which needs an object of type *Food.Event*, which is satisfied in (1c) because *the lunch* is of type *Lunch*, which is a subtype of *Food.Event*.

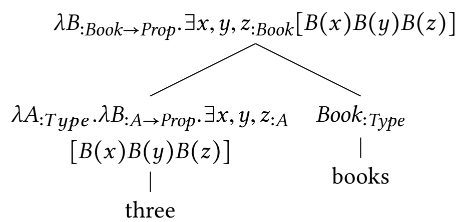
[4] As noticed by an anonymous reviewer, using types to model nouns prevents their predication. For example, the sentence "Iliad is a book", is then a type assertion, *Iliad : Book*, not a proposition.

(10) a.

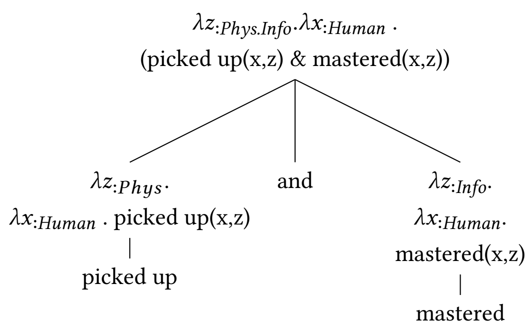


So far it seems that the notion of dot types along with the theory of coercive subtyping has been able to provide a solution for the case of copredication. Now, we will examine this approach for copredication sentences such as (4c) which includes quantification with distinct criteria of individuation. In (4c) *picked up* takes an argument of type *Physical*, whereas *mastered* needs an argument of type *Informational*. Assuming quantifier raising just as we did in the previous section, the analysis for (4c) proceeds as follows:

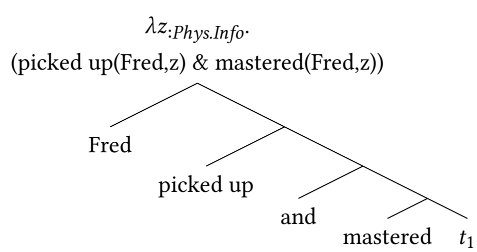
(11) a.



b.



c.



As can be seen in (11a), *three books* needs an argument of type *Book* → *Prop* as its input, and *Fred picked up and mastered* t_1 in (11c) provides an object of type *Physical.Informational* → *Prop*. Considering the covariance relationships similar to what we have in (9), we know that the latter is a subtype of the former, and thus the composition proceeds as in (12).

(12) a.

$\exists x, y, z_{:Book}$ [picked up(Fred,x) and mastered(Fred,x)
picked up(Fred,y) and mastered (Fred,y)
picked up(Fred,z) and mastered (Fred,z)]

$\lambda B_{:Book \to Prop}$.
$\exists x, y, z_{:Book}$ [B(x) B(y) B(z)]
three books

$\lambda z_{:Phys.Info}$.
(picked up(Fred,z) and
mastered(Fred,z))
Fred picked up and
mastered t_1

In the analysis shown in (12), quantification is still carried out over objects of type Book, because it returns the truth-condition of (4c) as follows: the sentence is true iff there are three books x, y, z which Fred picked up and mastered. Including the notion of coercive subtyping, however, we have the following coercive relations:

(13) a. $Book <_{c_1} Physical$

b. $Book <_{c_2} Informational$

which enable us to replace *picked up (Fred,x)* in the final analysis of (12) with *picked up (Fred, c_1(x))* and similarly *mastered (Fred,x)* with *mastered (Fred, c_2(x))*. So the truth-condition shown in (12) can be reformulated as follows:

(14) $\exists x, y, z : Book$
$[picked\ up(Fred, c_1(x))\ and\ mastered(Fred, c_2(x))$
$picked\ up(Fred, c_1(y))\ and\ mastered(Fred, c_2(y))$
$picked\ up(Fred, c_1(z))\ and\ mastered(Fred, c_2(z))]$

As we already mentioned in previous sections, intuitively we judge that (4c) is true iff the three books picked up and mastered by Fred are both physically and informationally distinct. The truth-condition formulated in (14), however, does not agree with this intuition unless we add an additional axiom to the system, as Chatzikyriakidis et al. [Chatzikyriakidis and Luo 2015] also point out in a footnote (fn.9); in coercive subtyping in general, when $X <_c Y$, the proposition $x \neq_X y$ does not entail $x \neq_Y y$ unless c is injective. Applying this general rule to (14) then, having three distinct objects of type *Book*, i.e., x, y, z does not entail that the corresponding coerced objects in *Physical*, i.e., $c_1(x)$, $c_1(y)$ and $c_1(z)$ are also distinct. Chatzikyriakidis et al. therefore, state that they axiomatically assume so for the atomic types like *Book* and *Physical*. While this assumption drives the desired meaning for (14), it creates some other problems: If c_2 is injective, then there cannot be two distinct books whose content is the same. Let x, y be two copies of the same book. So since $c_2(x) = c_2(y)$ and c_2 is injective, we will have $x = y$. Since c_1 is also injective, we will have $c_1(x) = c_1(y)$ and therefore the books are physically the same which is a contradiction.

A similar approach can be found in [Bekki and Asher 2012] where a subtype relation is represented as an injection sending an element of a type to itself which is then regarded as an element of its supertype. For polysemous common noun it is assumed that a number of aspect functions exist such that each one send an object of a common noun to one of its aspects:

$$asp_I : Book \rightarrow Info$$
$$asp_P : Book \rightarrow PhyObj$$

Aspect functions are different from injections of subtype relations. When encountering a type mismatch, the type of the functions is being shift, instead of the type of nouns (which are arguments to the functions). Aspects are used to shift the type functions when it is needed: *pickup* is the term $\lambda y \lambda x.\ pickup(x, y)$ of type *Physical* → *Animate* → *Prop* which by by the CCG-style functional composition with the aspect function asp_P shifts to the term $\lambda y \lambda x.\ pickup(x, asp_P(y))$ of type *Book* → *Animate* → *Prop*. As a result, the denotation of the sentence *Fred picked up and mastered a book* will be the term $\exists y$[5] (*book(y)* ∧ *pickup(Fred,*$asp_P(y)$) ∧ *master(Fred,*$asp_I(y)$)). [Bekki and Asher 2012] does not mention how it might handle the individuation problem. If the denotation for *three* is similar to that of *a*, then we will probably end up having something like (15):

(15) $\exists x, y, z$
[*book(x)* ∧ pickup(*Fred,* $asp_P(x)$) ∧ *mastered (Fred,* $asp_I(x)$)
book(y) ∧ pickup(*Fred,* $asp_P(y)$) ∧ *mastered (Fred,* $asp_I(y)$)
book(z) ∧ pickup(*Fred,* $asp_P(z)$) ∧ *mastered (Fred,* $asp_I(z)$)]

The result however has the same inadequacy as (14). Here we assert that there are three distinct objects of type *book* such that Fred picked up their physical aspects and mastered their content. But whether their physical aspects are distinct or their content are distinct is remained unspecified resulting problems we alluded to before.

5 COPREDICATION IN THE MONTAGOVIAN GENERATIVE LEXICON

[Retoré 2014] uses many-sorted higher order predicate calculus for semantic representation where the sorts are the base types. Meaning assembly is done using second order λ-calculus (Girard system F) as opposed to Luo's use of type theory with coercive subtyping. A a word in the lexicon is associated with a a finite set of λ-terms, one of them called the principal ?-term, the other ones are called optional. Words and constituents compose using the functional application rule. Semantic incompatibility is modeled by type mismatch as before. To allow an a priori type mismatch where it is legitimate, the optional λ-terms are used.The optional terms, change the type of the function or the argument under composition to resolve the type mismatch. To allow or block felicitous and infelicitous copredications, optional λ-terms are tagged as rigid or flexible. Flexible terms can be used without any restriction. Rigid terms cannot be used in copredication sentences.

Consider the following example:

(16) a. Liverpool is spread out.
b. Liverpool voted.
c. Liverpool won.
d. Liverpool is spread and voted last Sunday.
e. # Liverpool voted and won last Sunday.

Assuming the base types are defined as follows:

$$F : \text{football team} \quad T : \text{town} \quad P : \text{people} \quad Pl : \text{place}$$

[5] [Bekki and Asher 2012] do not specify the type of the existential variable y here. Our guess is that the type of this variable is the type *Entity* which is a supertype of *Book*.

Assume the principal term for *Liverpool* is a term of type T and its optional λ-terms defined as:

$$t_1 : T \to F \quad \text{rigid}$$
$$t_2 : T \to P \quad \text{flexible}$$
$$t_3 : T \to Pl \quad \text{flexible}$$

Now t_3 is used to resolve the type mismatch in (16a), t_2 for (16b), etc. The flexibility of both t_2 and t_3 give an account for the felicity of (16d) and the rigidity of t_1 handles the infelicitous copredication in (16e).

[Retoré 2014] does not elaborate on the individuation problem in copredication sentences, so it is unclear to us if the setting, outlined in [Retoré 2014], has an account for handling individuation or not.

6 SEMANTIC INTERPRETATION IN HOMOTOPY TYPE THEORY

In this section we first summarize (in the form of a puzzle) our analysis of what we think is the main cause of the complexity of the quantified copredication phenomena. The puzzle is inspired by work of Gotham in [Gotham 2012][6]. According to him

> Existing attempts to devise theories that predict ontologically respectable paraphrases as the natural language interpretations of copredication sentences all face problems of greater or lesser severity when it comes to extending those theories to cases where the copredication sentences involve numeric quantification. The general issue is that copredication presents us with divergent criteria for identifying objects falling under the denotation of the noun, and hence potentially divergent criteria for individuating and counting those objects. Native speaker judgements about copredication sentences involving numeric quantification do not perfectly reflect the counting and individuation criteria that the existing accounts predict.

We then give an informal and intuitive account of our proposal. Then there comes a brief introduction to homotopy type theory in which we implement the proposal . The introduction to homotopy type theory is focussed on the aspects of the theory that we are directly using in our implementation, namely the observation that identity type equips each type with the structure of a (weak) ω-groupoid, as studied in higher category theory. While this fact about the identity types is compatible with pure intensional type theory, we use homotopy type theory because the intuition behind our model comes from the homotopical interpretation of identity types, without which it is hard to express or justify the intuition.After the introductory section, we introduce our model and discuss how it deals with the copredication phenomenon.

6.1 The puzzle

To sum up our discussion so far, the puzzle comes from count nouns that are multi dimensional (that is, they feature more than one aspect) such that each dimension (aspect) has its own criteria of individuation. For example, the noun *book* has at least two dimensions: a physical dimension and an informational dimension. According to the individuation criteria of the first dimension, two books are distinct if they are physically distinct even if they are copies of the same book. According to the second dimension's criteria for individuation, two books are distinct if they are distinct content wise even if they are contained in one and the same volume. Now if the relation between *book* and its aspects is the relation between a whole and its parts then there are four possibilities for the criteria of individuation of books: 1) Books

[6]Gotham distiguishes between three approaches: 1) merelogical accounts (e.g., [Cooper 2007]). 2) Type Composition Logic (e.g., [Asher 2011]). 3)Pragmatic accounts involving lexical ambiguity (e.g., [Nunberg 2004])

are distinct if they are distinct physically. 2) Books are distinct if they are distinct content-wise. 3) Books are distinct if at least one of their parts are distinct. 4) Books are distinct just in case they are distinct in all their parts.However there are situations in which none of these possibilities work.

Consider the following sentence:

(17) Fred has five books on the shelf in his study room.

The following four cases can be considered:

(1) If distinct books are those that are physically distinct (that is, if the criteria of individuation of books is based on their physical aspect) then (17) means that there are five physically distinct books on the shelf. Now assume the situation where we have the following books on the shelf: two copies of Euclid's Elements, Divine Comedy, Iliad and Odyssey. Note that this list of books is compatible with (17) and the criteria of individuation under consideration. Assume that Fred read and mastered all the books on the shelf. Now consider the following sentence:

 (18) Fred mastered five books.

 So this case would judge (18) true because there *are* five books on the shelf. But intuitively (18) is true if and only if there are five informationally distinct books on the shelf. This indicates that the assumption that books are individualized physically is a wrong analysis.

(2) For the case where the criteria of individuation of books is based on their informational aspect consider the following scenario: Euclid's Elements, Divine Comedy and The Goldsworthy Trilogy[7] (informationally three books but physically only one). The trilogy counts as three books because in this case the criteria of individuation is based on informational aspect of books. Now consider the following sentence:

 (19) Fred picked up five books.

 So this case would judge (19) true which intuitively is not the case.

(3) Suppose distinct books are those that are either informationally distinct or physically distinct. This case also would judge (19) true for the scenario of the previous case.

(4) Suppose distinct books are those that are both informationally distinct and physically distinct (in other words, books are distinct whenever all of their parts are distinct). Now consider the situation where we have the following books on the shelf: Euclid's Elements, Divine Comedy, Iliad and two copies of Odyssey. This case would judge (19) false because according to analysis under consideration in this case, there *are not* five books on the shelf.

6.2 Hypothesis

Quantified multidimensional count nouns, semantically speaking, contain all possible semantic possibilities (in other words, they are ambiguous but not in an arbitrary way). When a quantified count noun passes as an argument to a predicate, its semantic value becomes crystalized by shrinking to a subset[8] of its possibilities (potentialities). It stays crystalized in the remaining context unless it is passed as an argument to another predicate where it will undergo further crystallization (that is, it shrinks to a yet (possibly) smaller subset of its possibilities).

[7] a collection of three books in one book

[8] Depending on the predicate this subset can be a proper subset or it can be equal to the original set of possibilities.

When I hear "three books", what comes into my mind is a range of possibilities: It can be three informationally and physically distinct books, two copy of the same book and a third informationally distinct book, a trilogy, ... Then if I hear "...picked up three books", the range of the possibilities shrink to those in which there are three physically distinct books. Moreover if I hear "...picked up and mastered three books", the collection of possibilities shrink more to those in which there are three physically and informationally distinct books. Similarly if I hear "three easy to understand books" or "three heavy books" the collection of possibilities shrink. But if I hear "...bought three books", the set of possibilities remains the same.

6.3 Identity type in homotopy type theory

In intensional Martin-Lof type theory, if a and b are objects of type A then the identity type $Id(a, b)$ (or equivalently $a = b$) is a proposition, namely, the proposition that a and b are identical. [Martin-Löf 1998]. Whether any two elements of an identity type are necessarily equal is a property called UIP (Uniqueness of Identity Proofs). [Coquand 1992] and [Altenkirch et al. 1994] showed that UIP is derivable in a type theory augmented with pattern matching but [Hofmann and Streicher 1998] proved that UIP is not derivable in pure type theory[9]. They refute the principle of UIP by providing a counter model interpreting types as groupoids[10] where an element of a type is an object of a groupoid and the proof of an equality between two elements is a morphism in the groupoid. Groupoid model shows that we can have multiple different proofs of the same identity. Wether the proofs of equality between proofs can also be multiple is a question which groupoid model even fails to express because in the groupoid structure the notion of morphisms between morphisms is absent. Homotopy type theory ([Awodey and Warren 2009] ; [Voevodsky 2011]) is based on this observation that not only we can have the proof relevant notion of equality between elements but also the proof relevant notion of equality between proofs of equality and we can continue this proof relevancy of equalities up to infinity [11].

A type with exactly one element (up to equality[12]) is called contractible. We say that a type is a proposition if it has at most one inhabitant[13]. A type whose equalities are propositions is called a set. It is known that any type with a decidable equality is a set [Hedberg 1998]. We say a type is a groupoid if all its equalities are sets. To continue with this hierarchy we say contractible types are of dimension (also called truncation level) [14] -2, propositions are of dimension -1, sets are of dimension 0 and groupoids are of dimension 1. A type has dimension $n + 1$ if its equalities are of dimension n. That is, a type A has dimension $n + 1$ if for all x, y in A, the type $x = y$ has dimension n. We say a type A is a n-type if A has dimension n. It is known that we can construct types which are not n-type for any natural number n [Kraus 2014].

6.4 The model

We call a type that is not n-type for any n, an *ideal* type. We postulate the existence of a universe, U_3[15] that is itself ideal and all of its elements are also ideal and that the following holds: for any equality type (of any order) in U_3, the equality is either empty or ideal.

[9] Independently, [Lamarche 1991] observed that type theory can be considered as an internal language of the category of groupoids.

[10] A groupoid is a category in which every morphism is an isomorphism.

[11] Such a structure is called a ω-groupoid which in homotopy theory has a model called Kan complex. The model, however, is using the axiom of choice. A constructive alternative model based on cubical sets was proposed in [Cohen et al. 2016]

[12] That is, a type A is contractible if it has an element a such that for all x in A we have $x = a$.

[13] An element of a type has many names: point, element, proof, token, witness or inhabitant.

[14] To be compatible with equivalent notions in homotopy theory we start counting with -2.

[15] We choose the index three because number three is associated with the meaning of multiplicity and it is this universe that contains and provides semantic meanings for lexical items.

The semantic denotation of a count noun is an identity type in U_3. The semantic denotation of an object (instance) of a count noun is a *nontrivial*[16] element of its corresponding type. For a count noun and each of its aspects there are types in U_3 which we call their prototypes. The semantic denotation of a count noun, then, is the equality between the prototype of the count noun and the prototypes of its aspects. For example, consider the common noun *book*. If we envisage that it has two aspects, namely physical and informational, then the semantic denotation of the common noun book, $[\![\text{Book}]\!]$, is defined as

$$\begin{aligned}(BookPrototype = PhysicalPrototype) \quad &= \\ (BookPrototype = InformationalPrototype)\end{aligned}$$

where the types BookPrototype, PhysicalPrototype and InformationalPrototype are in U_3. The semantic denotation of a book object is defined to be a *non trivial* element of $[\![\text{Book}]\!]$. So for example, $[\![\text{Iliad}]\!]$ is a *non trivial* element of $[\![\text{Book}]\!]$. Note that this definition has an interesting consequence: The existence of a particular book object, for example the book Iliad, entails that $[\![\text{Book}]\!]$ is ideal[17]. One may argue that this definition of the common noun book is strange or unreasonable because the type of book cannot be equated to the type of physical for the simple reason that books and physical things are not the same. To reply we refer to a popular example in topology: a coffee mug and a donut are the same as far as their topology is concerned. That is, there exists a homeomorphism between the surfaces of a donut and a coffee mug (with one handle).In other words, two spaces are homeomorphic if one can be deformed into the other by a continuous deformation without using cutting or glueing. If two spaces are homeomorphic, their topological properties will be identical, and therefore they are considered topologically the same. Now we cannot eat coffee mug nor we can drink tea using a donut instead of a mug. Likewise the type of *BookPrototype* can be continuously deformed into the type of *PhysicalPrototype*. We interpret "continuously", to mean without stopping being essentially what it was. On the on hand, when a book considered as its physical aspect, this consideration is total, in the sense that we treat the word as if it is a different word with no other aspect or sense. On the other hand, we do not exclude the fact that the physical thing under consideration is a book and indeed at any moment it can be turned into a book again[18]. The continuity of the deformation can be interpreted to reflect the latter consideration.

The semantic denotation of the physicality, $[\![\text{Physical}]\!]$, is defined as

$$(PhysicalPrototype = PhysicalPrototype)$$

The semantic denotation of a physical object is a *non trivial* element of $[\![\text{Physical}]\!]$. The non triviality assumption is mathematically crucial as we demonstrate later in this section. But one may ask what is the semantic significance of this assumption. We interpret the non triviality requirement as the following: a physical object is a particular deformation of the type PhysicalPrototype into itself. Now a physical object is temporary, that is, it has a start and an end. Its start coincides with the start of the deformation and its end with the end of deformation. In a trivial deformation the start and the end is the same and therefore a physical object corresponding to this deformation has no temporal existence which is contradictory to the nature of physical things.

Similar to the denotation of the physicality, we define the semantic of the informational, $[\![\text{Informational}]\!]$, as

$$(InformationalPrototype = InformationalPrototype)$$

[16] This consideration plays an important role in the consistency of our model. We elaborate on this later in this section.

[17] That is, the existence of the book Iliad affirms the existence of other books which are different from Iliad. Or if we consider *geometry* as a whole to be a type, then the existence of euclidean geometry affirms the existence of non-euclidean ones.

[18] For example in the sentence "I picked up Iliad and read it", the book Iliad is a physical thing under the predicate "pick up", but it will turn again into a book to become an informational object when it is under the predicate "read".

Considering the definition of $[\![\text{Book}]\!]$, one may object that as soon as you have an object b of type *Book*, you will have its inverse b^{-1}. Now what is the semantic interpretation of b^{-1}? We reply that b and b^{-1} refer to the same book but the accentuation is different. Consider the following sentences:

(20) a. Iliad is heavy[19] but easy to understand.

b. Iliad is easy to understand but heavy.

If b is the meaning of Iliad in the first sentence then b^{-1} would be its meaning in the second sentence. What happens in mind when constructing b, is first a deformation of BookPrototype to PhysicalPrototype and then adding its informational component. Whereas for b^{-1} it is the morphing of BookPrototype into InformationalPrototype that is happening first.

Now we need to show that if we have a book object, we will have a physical object and an informational object such that the physical object can turn into a book object at any moment it is required to do so. Likewise for the informational object. Furthermore we need to show that the proposed definition of $[\![\text{Book}]\!]$ affords the fluidity that our hypothesis requires. That is when we say "three books", the semantic denotation should afford all the possible meanings of this utterance[20]. Consider there is a book object with its semantic denotation denoted by b. By definition b is a nontrivial element of $[\![\text{Book}]\!]$[21]. The non-triviality of b entails that none of the types BookPrototype = PhysicalPrototype and BookPrototype = InformationalPrototype is empty. Now an equality in U_3 is either empty or ideal, therefore both equality types are ideal. The element b induces an equivalence function, namely f, from BookPrototype = PhysicalPrototype to BookPrototype = InformationalPrototype ([Univalent Foundations Program 2013] Lemma 2.10.1). For the element b we postulates the existence of two elements b_p : (BookPrototype = PhysicalPrototype) and b_i : (BookPrototype = InformationalPrototype). We define the physical object corresponding to the book object b, calling it *b qua physical*, as the following:

(21) $b \text{ } qua \text{ } physical = b_p^{-1} \circ f(b_p) \circ b_i^{-1} \circ f^{-1}(b_i)$.

So *b qua physical* is a concatenation of four paths:[22] The element b_p is a path form BookPrototype to PhysicalPrototype so its inverse is a path from PhysicalPrototype to BookPrototype. The element $f(b_p)$ is a path from BookPrototype to InformationalPrototype. The element b_i^{-1} is a path from InformationalPrototype to BookPrototype and finally the element $f^{-1}(b_i)$ is a path from BookPrototype to PhysicalPrototype. So by concatenating these four paths we construct a path from PhysicalPrototype to itself and therefore we construct *b qua physical* which is an element of $[\![\text{physical}]\!]$. We believe this definition is determined enough to tie *b qua physical* to b meaningfully and fluid enough to give freedom to *b qua physical* to be equal to some other *c qua physical* for a book object c.

Similarly we define the informational object corresponding to the book object b, calling it *b qua informational*, as the following:

(22) $b \text{ } qua \text{ } Informational = b_i^{-1} \circ f^{-1}(b_i) \circ b_p^{-1} \circ f(b_p)$.

We define a sentence to be *interpretable* if and only if it has a type. The semantic denotation of an interpretable sentence is defined to be its type. Note that according to this definition the semantic value of a sentence is a proof

[19] Here heavy is intended to mean physically heavy and not metaphorically heavy which would be contradictory to the latter part of the sentence.
[20] This utterance may mean three physically distinct books or a trilogy or three copies of the same book etc.
[21] This very fact asserts that $[\![\text{Book}]\!]$ is ideal.
[22] Inspired by the homotopical interpretation of types, elements of an equality type are sometimes called paths. So here by concatenating four paths we are constructing an element of $[\![\text{Physical}]\!]$.

relevant concept. That is, we believe natural language sentences, semantically speaking, are not mere propositions which are either true of false. Rather the meaning of a sentence is a witness, a proof or an evidence (all terms refer to the same thing), or a collection of evidence that the sentence is true. A sentence is meaningless if there is no evidence, no intuition, no proof or nothing that asserts its truth. The things that asserts the truth of a sentence are its denotational meaning. Whether it is one evidence or several ones, proving the truth of the sentence, is the same as far as the truth of the sentence is concerned. Nevertheless as far as the meaning of the sentence is concerned, all those evidence count toward the meaning of the sentence. If I say "Fred is 30 years old", the meaning of this sentence is all the evidence that assert the truth of Fred being of this age. I may have only one of those evidence, which is enough for me to believe that the sentence is true but the meaning of the sentence is all the evidence attesting to its truth.

Consider the following sentences:

(23) a. Fred picked up Iliad.
b. Fred mastered Iliad.
c. Fred picked up and mastered Iliad.

We define the denotation of the predicate[23] in (23a) as

$$[\![\text{pick up}]\!] = \lambda x : [\![\text{Human}]\!]\ \lambda y : [\![\text{Physical}]\!]\ .\text{type of all the evidence}^{24} \text{ that } x \text{ picked up } y.$$

We use the notation PICKUP(x, y) to mean the type of all the evidence that x picked up y. We insist on repeating the redundant phrase "all of the evidence" to accentuate the fact that the type under consideration is not a proposal, nor the sentence describing the type is a truth condition. Similarly we use the notation MASTER(x, y) to mean the type of all the evidence that x mastered y.

Now assuming $[\![\text{Fred}]\!] = h : [\![\text{Human}]\!]$ and $[\![\text{Iliad}]\!] = b : [\![\text{Book}]\!]$, the semantic value of (23a) is computed as

$$[\![(23a)]\!] = [\![\text{pick up}]\!]\ (h)(b\ \textit{qua physical}) = \text{PICKUP}(h, b\ \textit{qua physical})$$

Likewise the denotation of the predicate in (23b) is defined as

$$[\![\text{master}]\!] = x : [\![\text{Human}]\!]\ \lambda y : [\![\text{Informational}]\!]\ .\text{MASTER}(x, y).$$

So the semantic value of (23b) is computed as

$$[\![(23b)]\!] = [\![\text{master}]\!]\ (h)(b\ \textit{qua informational}) = \text{MASTER}(h, b\ \textit{qua informational})$$

As for (23c), we first define the semantic denotation of *and* as

$$[\![\text{and}]\!] = \lambda A\,B.\Sigma AB.$$

Where A and B are types. Note that ΣXY is true if and only if there exists an element x in X for which $Y(x)$ is inhabited. If Y is not dependent on type X, then $Y(x)$ simply means Y. So in a case where Y is not dependent on X, the type ΣXY is inhabited only if both types X, Y are inhabited. The semantic value of (23c) is computed as

$$\begin{aligned}[\![(23c)]\!] &= \Sigma([\![\text{pick up}]\!]\ (h)(b\ \textit{qua physical}))\ ([\![\text{master}]\!]\ (h)(b\ \textit{qua informational}))\\ &= \Sigma\ \text{PICKUP}(h, b\ \textit{qua physical})\ \text{MASTER}(h, b\ \textit{qua informational})\end{aligned}$$

Now consider the following sentences:

(24) a. Fred picked up three books.
b. Fred mastered three books.

[23] We disregard the tense of the verb as it is orthogonal to our analysis.

c. Fred picked up and mastered three books.

The semantic value of (24a) is computed as

⟦(24a)⟧ = The type of all the evidence that there exist x, y, z in ⟦Book⟧ such that (the type of all of the evidence that x *qua physical* and y *qua physical* and z *qua physical* are distinct and Σ(PICKUP(h, x *qua physical*)) (Σ(PICKUP (h, y *qua physical*)) (PICKUP (h, z *qua physical*))))[25]

Likewise for (24b) we have the following

⟦(24b)⟧ = The type of all the evidence that there exist x, y, z in ⟦Book⟧ such that (the type of all the evidence that x *qua informational* and y *qua informational* and z *qua informational* are distinct and Σ(MASTER(h, x *qua informational*)) (Σ(MASTER (h, y *qua informational*)) (MASTER (h, z *qua informational*))))

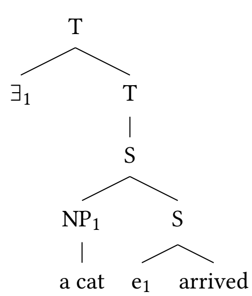
And finally the semantic value of (24c) is computed as

⟦(24c)⟧ = The type of all the evidence that there exist x, y, z in ⟦Book⟧ such that Σ (the type of all of the evidence that x *qua physical* and y *qua physical* and z *qua physical* are distinct and Σ(PICKUP(h, x *qua physical*)) (Σ(PICKUP (h, y *qua physical*)) (PICKUP (h, z *qua physical*)))) (the type of all the evidence that x *qua informational* and y *qua informational* and z *qua informational* are distinct and Σ(MASTER(h, x *qua informational*)) (Σ(MASTER (h, y *qua informational*)) (MASTER (h, z *qua informational*))))

6.5 Computation details

In her analysis of indefinites (e.g., *a book*), [Heim 1982] takes indefinites as variables and assumes a covert existential quantifier which scopes over the entire sentence and unselectively binds the indefinites in an un-embedded sentence (through the rule of *Existential Closure*). For example, for the sentence *A cat arrived* , she assumes the structure shown in (25) as the logical form: first *a cat* is moved out of its phrase through the rule of NP Prefixing, leaving behind e_1. Then $\exists_1$ adjoins to the top node, and binds the indefinite *a cat*:

(25)



We extend this analysis to numbered NPs (e.g., *three books*) in the following way: in the syntactic tree of the sentence *Fred picked up three books*, for example, we assume that *three books* is moved out of its phrase, adjoining to the S node as $three_d$ *books* (d for definite). But in addition to $three_d$ *books*, we assume that there is also $three_i$ *books* (i for indefinite) adjoined to the top node with a similar function as that of the covert existential quantifier in [Heim 1982]'s analysis.

The semantic denotation of $three_i$ is therefore defined as:

⟦$three_i$⟧ = $\lambda A_{:U_3} \lambda D$. The type of all the evidence that there exists x, y, z in A such that D, where D is a type containing x, y and z as free variables of type A.

[25] When we say "…such that (the type of all of the evidence that …)", we mean to express the condition that "…such that (the type (of all of the evidence that …) is not empty.)".

the semantic denotation of $three_d$ is defined as below:

$[\![\text{three}_d]\!] = \lambda A{:}U_3 \lambda B{:}C \to U_3$. The type of all the evidence that that x *qua* C and y *qua* C and z *qua* C are distinct and $[\![\text{and}]\!]\ (B(x\ qua\ C))([\![\text{and}]\!]\ (B(y\ qua\ C))(B(z\ qua\ C))$, where x, y and z are free variables of type A.

(1) Computation of $[\![(24a)]\!]$

The semantic value of *Fred picked up* t_1 is computed as follows:

$[\![\text{Fred picked up t}_1]\!] = \lambda x : [\![\text{Physical}]\!]$. PICKUP(h, x).

Then, we calculate the denotation of $three_d$ *books*.

$[\![\text{three}_d \text{ books}]\!] = \lambda B{:}C \to U_3$. The type of all the evidence that x *qua* C and y *qua* C and z *qua* C are distinct and $[\![\text{and}]\!]\ (B(x))([\![\text{and}]\!]\ (B(y))(B(z))$, where x, y and z are free variables of type $[\![\text{book}]\!]$.

Passing the denotation of *Fred picked up* t_1 to the denotation of $three_d$ *books*, we get the following:

$[\![[\text{three}_d \text{ books [Fred [picked up t}_1]]]]\!]$ = The type of all the evidence that x *qua physical* and y *qua physical* and z *qua physical* are distinct and $[\![\text{and}]\!]$ (PICKUP(h, x *qua physical*))($[\![\text{and}]\!]$ (PICKUP(h, y *qua physical*))(PICKUP(h, z *qua physical*)), where x, y and z are free variables of type $[\![\text{book}]\!]$.

For future reference, we use $P[x, y, z]$ to refer to $[\![[\text{three}_d \text{ books [Fred [picked up t}_1]]]]\!]$.

Finally we have:

$[\![[24a]]\!]$ = The type of all the evidence that there exist x, y, z in $[\![\text{Book}]\!]$ such that (the type of all of the evidence that x *qua physical* and y *qua physical* and z *qua physical* are distinct and Σ(PICKUP(h, x *qua physical*)) (Σ(PICKUP (h, y *qua physical*)) (PICKUP (h, z *qua physical*)))).

(2) Similarly for (24b) we have:

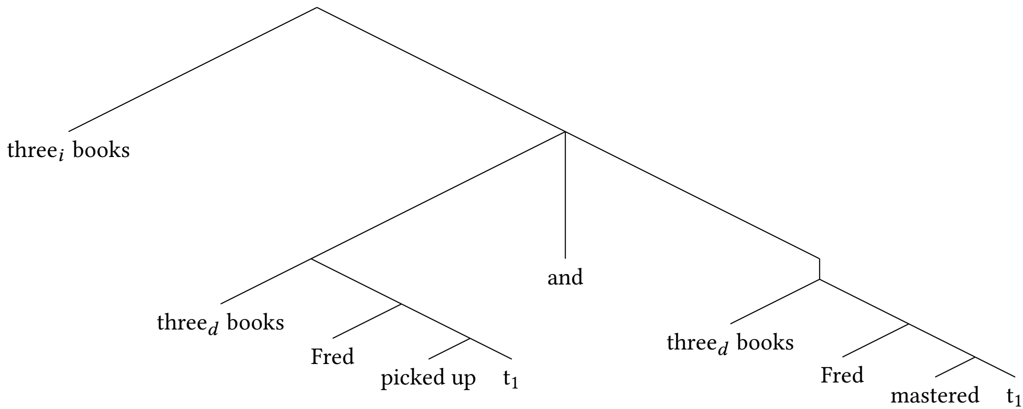
$[\![[\text{three}_d \text{ books [Fred [mastered t}_1]]]]\!]$ = The type of all the evidence that x *qua informational* and y *qua informational* and z *qua informational* are distinct and $[\![\text{and}]\!]$ (MASTER(h, x *qua informational*))($[\![\text{and}]\!]$ (MASTER(h, y *qua informational*))(MASTER(h, z *qua informational*)), where x, y and z are free variables of type $[\![\text{book}]\!]$.

For future reference, we refer to $[\![[\text{three}_d \text{ books [Fred [mastered t}_1]]]]\!]$ as $M[x, y, z]$.

$[\![[(24b)]]\!]$ = The type of all the evidence that there exist x, y, z in $[\![\text{Book}]\!]$ such that (the type of all of the evidence that x *qua informational* and y *qua informational* and z *qua informational* are distinct and Σ(MASTER(h, x *qua informational*)) (Σ(MASTER (h, y *qua informational*)) (MASTER (h, z *qua informational*)))).

(3) Computation of $[\![(24c)]\!]$

The syntactic structure of (24c) is as follows:



Now we can compose the meanings computed above for (24a) and (24b) to get the semantic value of (24c):
To make the following computations more readable, we call the subtree to the right of *and*, tree$_a$ and the one on the left, tree$_b$.

$[\![\text{tree}_b \text{ and tree}_a]\!] = \Sigma\, [\![\text{tree}_a]\!]\, [\![\text{tree}_b]\!] = \Sigma(P[x,y,z])\ (M[x,y,z])$

$[\![(24c)]\!] = [\![\text{three}_i \text{ books}]\!]\ ([\![\text{tree}_b \text{ and tree}_a]\!])$ = The type of all the evidence that there exist x, y, z in $[\![\text{Book}]\!]$ such that ($[\![\text{tree}_b \text{ and tree}_a]\!]$)
= The type of all the evidence that there exist x, y, z in $[\![\text{Book}]\!]$
such that $(\Sigma(P[x,y,z])(M[x,y,z]))$
= The type of all the evidence that there exist x, y, z in $[\![\text{Book}]\!]$ such that Σ (the type of all of the evidence that x *qua physical* and y *qua physical* and z *qua physical* are distinct and Σ(PICKUP(h, x *qua physical*)) (Σ(PICKUP (h, y *qua physical*)) (PICKUP (h, z *qua physical*)))) (the type of all the evidence that x *qua informational* and y *qua informational* and z *qua informational* are distinct and Σ(MASTER(h, x *qua informational*)) (Σ(MASTER (h, y *qua informational*)) (MASTER (h, z *qua informational*))))

6.6 Comparison

The inadequacy of approaches in [Chatzikyriakidis and Luo 2015] and [Bekki and Asher 2012] can be resolved if they use the rule of existential closure as it is used in the analysis we just gave. That is, for copredication sentences, if the same syntactic structure as in section (6.5) is used, then in terms of adequacy there is no difference between [Chatzikyriakidis and Luo 2015], [Bekki and Asher 2012] and our approach. The advantage, however can be best described by the following remark in [Šimon and Huang 2010]:

> …Intuitively, there is another problem with the notion of complex argument as a product from which the simple constituent types can be retrieved via projections. Take for example the word *book*: the theories coercing *book*$^{P.I}$ into either *book*^{P} or *book*^{I} loose an important aspect of the meaning of book, which becomes either a bare "physical object" or a bare "information". In order to talk about meaning of book, both components have to be present. We can manipulate with books the same way as we do with some general physical object, we can for example carry, drop or throw them by the virtue of them being subtypes of "physical object" and we can formalize this neatly in logic or a type theory. But where does the rest of

> the meaning of book go? Objects can be manipulated by casting their type into an appropriate type or, as we want to argue here, by virtue of types that constitute that object. In other words, we need a notion of structured meaning. Objects can be "transformed" or viewed from different perspective without loosing any of their meaning components. We are arguing against casting more complex types into simpler ones and loosing information in the process.

As it is discussed in section (6.4), an object b of type book can be viewed from different perspectives as:

(26) a. b *qua physical* $= b_p^{-1} \circ f(b_p) \circ b_i^{-1} \circ f^{-1}(b_i)$.
b. b *qua Informational* $= b_i^{-1} \circ f^{-1}(b_i) \circ b_p^{-1} \circ f(b_p)$.

Therefore b *qua physical* and b *qua Informational* do not loose any of the meaning components of b. The difference between the two "transformations" is precisely the difference between the perspectives, reflected here in the order of the components in (26).

7 CONCLUSION

In pure intensional type theory, identity types need not be subject to UIP. This fact together with a homotopical interpretation of identity types in homotopy type theory provides a justification for modeling common nouns as identifications of their aspects. We showed that this model, while being simple, is able to successfully handle a semantically complex phenomenon, namely copredication when it involves individuation. Our solution relies on an analysis of numerical quantifiers which is inspired by Heim's treatment of indefinites, and a new approach to meaning in formal semantics, which establishes that the denotation of a semantically interpretable sentence is a type, and the sentence is true iff its type is not empty. The type of a sentence is then envisaged as the collection of all the proofs, witnesses or evidence that the sentence is true. We showed that in this framework, copredication can be modeled with no need to subtypes and the complexities they bring about.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1217454. For useful comments and insightful criticism, we are grateful to Thomas Grano, the audience at the 4th workshop on Natural Language and Computer Science at Columbia University and the anonymous reviewers of TYPES post-proceedings.

REFERENCES

Thorsten Altenkirch, Veronica Gaspes, Bengt Nordström, and Björn von Sydow. 1994. A userâĂŹs guide to ALF. *Chalmers University of Technology, Sweden* (1994).

Nicholas Asher. 2008. A type driven theory of predication with complex types. *Fundamenta Informaticae* 84, 2 (2008), 151–183.

Nicholas Asher. 2011. *Lexical meaning in context: A web of words*. Cambridge University Press.

Steve Awodey and Michael A Warren. 2009. Homotopy theoretic models of identity types. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 146. Cambridge University Press, 45–55.

Daisuke Bekki and Nicholas Asher. 2012. Logical polysemy and subtyping. In *JSAI International Symposium on Artificial Intelligence*. Springer, 17–24.

Stergios Chatzikyriakidis and Zhaohui Luo. 2013. An account of natural language coordination in type theory with coercive subtyping. In *Constraint Solving and Language Processing*. Springer, 31–51.

Stergios Chatzikyriakidis and Zhaohui Luo. 2015. Individuation criteria, dot-types and copredication: A view from modern type theories. In *ACL anthology*.

Alonzo Church. 1940. A formulation of the simple theory of types. *The journal of symbolic logic* 5, 02 (1940), 56–68.

Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2016. Cubical type theory: a constructive interpretation of the univalence axiom. *arXiv preprint arXiv:1611.02108* (2016).

Robin Cooper. 2007. Copredication, dynamic generalized quantification and lexical innovation by coercion. In *Fourth International Workshop on Generative Approaches to the Lexicon*.

Robin Cooper. 2011. Copredication, quantification and frames. In *Logical aspects of computational linguistics*. Springer, 64–79.

Thierry Coquand. 1992. Pattern matching with dependent types. In *Proceedings of the Workshop on Types for Proofs and Programs*. 71–83.

Matthew Gotham. 2012. Numeric quantification in copredication? (2012).

Matthew Graham Haigh Gotham. 2014. *Copredication, quantification and individuation*. Ph.D. Dissertation. UCL (University College London).

Michael Hedberg. 1998. A coherence theorem for Martin-Löf's type theory. *Journal of Functional Programming* 8, 4 (1998), 413–436.

Irene Heim. 1982. The semantics of definite and indefinite noun phrases. (1982).

Irene Heim and Angelika Kratzer. 1998. *Semantics in generative grammar*. Blackwell publishers, Cambridge (Mass.), Oxford. http://opac.inria.fr/record=b1080204

Martin Hofmann and Thomas Streicher. 1998. The groupoid interpretation of type theory. *Twenty-five years of constructive type theory (Venice, 1995)* 36 (1998), 83–111.

Nicolai Kraus. 2014. The general universal property of the propositional truncation. *arXiv preprint arXiv:1411.2682* (2014).

François Lamarche. 1991. A proposal about foundations I. (1991).

Zhaohui Luo. 1994. *Computation and reasoning: a type theory for computer science*. Oxford University Press, Inc.

Zhaohui Luo. 1999. Coercive subtyping. *Journal of Logic and Computation* 9, 1 (1999), 105–130.

Zhaohui Luo. 2012a. Common nouns as types. In *Logical aspects of computational linguistics*. Springer, 173–185.

Zhaohui Luo. 2012b. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy* 35, 6 (2012), 491–513.

Per Martin-Löf. 1998. An intuitionistic theory of types. *Twenty-five years of constructive type theory* 36 (1998), 127–172.

Richard Montague. 1975. Formal philosophy. (1975).

Richard Moot and Christian Retoré. 2012. *The logic of categorial grammars: a deductive account of natural language syntax and semantics*. LNCS, Vol. 6850. Springer. http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-31554-1

Bengt Nordström, Kent Petersson, and Jan M Smith. 1990. *Programming in Martin-Löf's type theory*. Vol. 200. Oxford University Press Oxford.

Geoffrey Nunberg. 2004. The pragmatics of deferred interpretation. *The handbook of pragmatics* (2004), 344–364.

Barbara Partee. 1984. Compositionality. *Varieties of formal semantics* 3 (1984), 281–311.

James Pustejovsky. 1991. The generative lexicon. *Computational linguistics* 17, 4 (1991), 409–441.

Aarne Ranta. 1994. Type-theoretical grammar. (1994).

Christian Retoré. 2014. The Montagovian Generative Lexicon Lambda Tyn: a Type Theoretical Framework for Natural Language Semantics. In *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, Vol. 26. 202–229.

Petr Šimon and Chu-Ren Huang. 2010. Cross-sortal Predication and Polysemy. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 2010)*. 853–861.

Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics*. http://homotopytypetheory.org/book, Institute for Advanced Study.

Vladimir Voevodsky. 2011. Univalent foundations of mathematics. In *International Workshop on Logic, Language, Information, and Computation*. Springer, 4–4.

Arnold Zwicky and Jerrold Sadock. 1975. Ambiguity tests and how to fail them. *Syntax and semantics* 4, 1 (1975), 1–36.