Individuation Criteria, Dot-types and Copredication: A View from Modern Type Theories

Abstract

In this paper we revisit the issue of copredication from the perspective of modern type theories. Specifically, we look at: a) the counting properties of dot-types, and b) the case of a complex dot-type that has remained unsolved in the literature, i.e. that of *newspaper*. As regards a), we show that the account proposed in (Luo, 2010) for dot-types makes the correct predictions as regards counting. In order to verify this, we implement the account in the Coq proof-assistant and check that the desired inferences follow. Then, we look at the case of b), the case of a dot-type which is both resource and context sensitive. We propose a further resource sensitive version of the dot-type, in effect a linear dot-type. This along with local coercions can account for the behaviour attested.

1 Copredication: Dot Types and Individuation Criteria

One of the issues that should be taken care of when giving an account of co-predication, concerns cases of coordination like the one shown below:

(1) John picked up and mastered three books

In the above sentence, the CN book is used in its physical sense (PHY) with respect to the predicate picked up, while for the predicate mastered it is rather used in its informational content sense (INFO). A theory of co-predication should be able to take care of these facts. This is true for the account by means of the dot-types proposed by (Luo, 2010; Luo, 2012b). However, besides capturing this behaviour of dot objects, there is an additional property that has to be captured. The account provided must also make the correct predications as regards individuation and counting. Let us explain. Consider the following sentences:

(2) John picked up three books

(3) John mastered three books

(4) John picked up and mastered three books

The first example (2) is true in case John picked three distinct physical objects. Thus, it is compatible with a situation where John picked up three copies of the same book. (3) is true in case three distinct informational objects are mastered but does not impose any restrictions on whether these three informational objects should be different physical objects or not. To the contrary, (4) is only compatible with an interpretation where three distinct physical objects as well as three distinct informational objects are involved.[1]

Another issue pertaining to dot types concerns cases of what Retoré (2014) calls rigid and flexible coercions in co-predication cases. These cases in contrast to cases like *Book* where both senses can be coordinated, involve examples where if one of the senses is used the other one cannot be used anymore:

(5) Liverpool is spread out and voted (last Sunday).

[1] This is basically an issue of how complex objects, i.e. dot-types, are individuated and stems from the work of (Asher, 2008; Asher, 2011).

(6) # Liverpool voted and won (last Sunday).

Perhaps a better example for such cases is Pustejovsky's *newspaper* examples. The CN *newspaper* is associated with three senses: a) physical object, b) informational object and c) institution. It is a strange fact that whereas senses a) and b) can appear together in a coordinated structure, sense c) cannot appear with any of the other two (examples taken from (Antunes and Chaves, 2003)):

(7) # That newspaper is owned by a trust and is covered with coffee.

(8) # The newspaper fired the reporter and fell off the table.

(9) # John sued and ripped the newspaper.

Pustejovsky's proposal (Pustejovsky, 1995) to treat newspaper as a composite dot object does not explain the above facts. Likewise, the proposal of using (ordinary) dot-types in (Luo, 2010) has a similar problem: one would consider *newspaper* to be a subtype of the dot-type $\textsc{Inst} \bullet (\textsc{Phy} \bullet \textsc{Info})$, which would not disallow the above bad examples. The picture gets complicated in the light of examples like the following, in which it seems that the institutional sense can be used together with one of the two other senses in some cases:

(10) The newspaper you are reading is being sued by Mia.

As far as we know, no satisfactory account has been provided to these questions yet. In this paper, following earlier work on dot-types in MTTs (Luo, 2010; Luo, 2012b; Xue and Luo, 2012) and coordination (Chatzikyriakidis and Luo, 2012), we take up the challenge of providing an account that correctly predicts the individuation criteria in cases of co-predication while it furthermore provides a first look at capturing the behaviour of problematic cases like *newspaper*.

2 Formal Semantics in Modern Type Theories: a Brief Introduction

The term Modern Type Theories (MTTs) refers to type theories studied and developed within the tradition of Martin-Löf, which include predicative type theories such as Martin-Löf's type theory (Martin-Löf, 1984; Nordström and Petersson, 1990) and impredicative type theories such as CIC_p as implemented in the Coq proof assistant (The Coq team, 2007) and UTT (Luo, 1994). Linguistic semantics in Modern Type Theories (MTT-semantics for short) was first studied by Ranta in his pioneering work (Ranta, 1994).[2] It has been further developed in the last several years, including the crucial employment of the theory of coercive subtyping (Luo, 1999; Luo, Soloviev and Xue, 1984) among other developments and made MTT-semantics a viable and full-blown alternative to the traditional Montagovian framework. In this paper, we use one such a modern type theory, Luo's UTT with Coercive Subtyping (Luo, 1994; Luo, 1999), whose application to linguistic semantics was first discussed in (Luo, 2010).

In this section, we briefly discuss some of the most distinctive features of MTTs, specifically the ones most relevant to this paper.

2.1 Type Many-sortedness and CNs as Types

The domain of individuals in MTTs is multi-sorted and not single-sorted as in Church's simple type theory (Church, 1940). Instead of using one coarse-grained domain of entities, like it is done in the Montague Semantics (MS) (Montague, 1974), MTTs contain many types that allow one to make fine-grained distinctions between individuals and further use those different types to interpret subclasses of individuals. For example, one can find $John : [\![man]\!]$ and $Mary : [\![woman]\!]$, where $[\![man]\!]$ and $[\![woman]\!]$ are different types.

A further difference closely related to type many-sortedness concerns the interpretation of CNs. In MS, CNs are interpreted as predicates of type $e \rightarrow t$, whereas in MTTs CNs are interpreted as *types*. Thus, in MTTs, CNs *man*, *human*, *table* and *book* are interpreted as types $[\![man]\!]$, $[\![human]\!]$, $[\![table]\!]$ and $[\![book]\!]$, respectively. (Such types may be defined by means of type constructors such as Σ etc – see below.) Then, individuals are interpreted as being of one of the types used to interpret CNs. Such interpretations of CNs as types would not work

[2]Potentially, even further back, with the work of Sundholm (Sundholm, 1986; Sundholm, 1989), but Ranta (Ranta, 1994) was the first systematic study of formal semantics in a modern type theory.

without a proper subtyping mechanism that extends MTTs – coercive subtyping provides us with such a framework.[3]

2.2 Rich Typing

Type structures in MTTs are very rich. They can be used to represent collections of objects (or constructive sets, informally) in a model-theoretic sense, although they are syntactic entities in MTTs. Elaborating on the expressiveness of typing structures of MTTs, we briefly mention the following type structures:

- Dependent sum types (Σ-types $\Sigma(A, B)$ which have product types $A \times B$ as a special case). Σ-types have been used to interpret intersective and subsective adjectives without the need of resorting to meaning postulates. The inferences follow directly from typing (Ranta, 1994; Chatzikyriakidis and Luo, 2013). Note that subtyping is essential for the Σ-type to work (Luo, 2012b).

- Dependent product types (Π-types $\Pi(A, B)$, which have arrow-types $A \to B$ as a special case). These are basic dependent types that, together with universes (see below), provide polymorphism among other things. To give an example, verb modifying adverbs are typed by means of dependent Π-types (together with the universe CN of common nouns) (Luo, 2012b; Chatzikyriakidis, 2014).

- Disjoint union types ($A + B$). Disjoint union types have been proposed to give interpretations of privative adjectives (Chatzikyriakidis and Luo, 2013).

- Universes. These are types of types, basically collections of types. Typical examples of universes in MTT-semantics include, among others, the universe $Prop$ of logical propositions as found in impredicative type theories and the universe CN of (the interpretations of) common nouns (Luo, 2012b). Further uses of universes can be seen in (Chatzikyriakidis and Luo, 2012) where the universe $LType$ of all linguistic types is used in order to deal with coordination.

- Dot-types ($A \bullet B$). These are special types introduced to study co-predication (Luo, 2012b). It is worth mentioning that coercive subtyping is essentially employed in the formulation of dot-types.[4]

2.3 MTT-semantics is Both Proof-theoretic and Model-theoretic

It has been noted recently (Luo, 2014) that one of the advantages of MTT-semantics as compared to traditional Montagovian approaches is that MTT-semantics can be seen as being both model-theoretic and proof-theoretic. NL semantics can first be represented in an MTT in a model-theoretic way and then these semantic representations can be understood inferentially in a proof-theoretic way (Luo, 2014).

In particular, since MTTs are proof-theoretically specified, it is not surprising that many proof assistants implement MTTs. Perhaps, the most advanced of these proof-assistants is the Coq proof-assistant (The Coq team, 2007). Coq is a state-of-the-art proof assistant that has produced a number of impressive results. Some of these include a complete mechanized proof of the four colour theorem (Gonthier, 2005), the odd order theorem (Gonthier et al., 2013) as well as CompCert, a formally verified compiler for C (Leroy, 2013). Because Coq has a powerful reasoning ability and that it implements an MTT, a new avenue of research is opened up – to use Coq as an NL reasoner. This has been attempted in (Chatzikyriakidis and Luo, 2014a; Chatzikyriakidis and Luo, 2014b) with a number of promising results as regards NL inference. In this paper, we also exemplify the way proof-assistants can be used to help in checking the inferences that semantic accounts give rise to.

3 CNs as Types and Individuation Criteria

As already discussed in our introduction to MTTs, CNs are interpreted as types in MTTs. This proposal has also some nice consequences concerning

[3] See (Luo, 1999; Luo, Soloviev and Xue, 1984) for the formal details of coercive subtyping. Also see (Luo, 2012a) and the next section for further argumentation on interpreting CNs as types.

[4] See (Bassac et al., 2010) for another proposal of using coercions to deal with co-predication.

what Geach (1962) has called the criterion of identity, which is pretty much the individuation criterion that we have been referring to in this paper. Intuitively, a CN determines a concept that beside having a criterion of application to be employed to determine whether the concept applies to an object, it further involves a criterion of identity, to be employed to determine whether two objects of the concept are the same. It has been argued that CNs are distinctive in this as other lexical terms like verbs and adjectives do not have such criteria of identity (cf. the arguments in (Baker, 2003)). There seems to be a close link between the constructive notion of a set (Type) and criteria of identity/individuation. This is because, in constructive mathematics, a set is a 'preset', which involves its application criterion, together with an equality, which further gives its criterion of identity determining whether two objects of the set are the same (Bishop, 1967; Beeson, 1985). Modern type theories such as Martin-Löf's type theory (Martin-Löf, 1975; Martin-Löf, 1984) were originally developed for the formalisation of constructive mathematics, where each type is associated with such an equality or criterion of identity. The identification of CNs as types thus provides CNs their criteria of application and identity. We cannot go into the details of how this is to be achieved formally. but the interested reader is directed to (Luo, 2012a) for a detailed exposition of the CNs as Types idea.

In order to proceed, firstly we have to discuss the existing account of dot-types as this was given by (Luo, 2010; Luo, 2012b; Xue and Luo, 2012). Specifically, we have to see whether this account predicts the counting criteria correctly in examples like (4) repeated below:

(11) John picked up and mastered three books

As we have said, the only possible interpretation of (11) we receive is one where three distinct physical as well as informational objects are involved. The sentences cannot be interpreted as involving three distinct informational objects but one physical object or vice versa as involving three distinct physical objects but one informational object. The question is whether this account captures that. First of all, let us say something about coordination, since this would be needed in discussing the examples in a compositional manner. The approach we suggest for coordination, based on earlier work in (Chatzikyriakidis and Luo, 2012) involves a type universe of linguistic types, $LType$:[5]

(12) $\Pi A : LType.\ A \to A \to A$

As regards typing the above is a natural way to encode coordination. However, we need a way to further encode the semantics of coordination in each case. For this paper, we show this for VP coordination only. In order to define VP coordination, we first define an auxiliary object AND:

(13) $AND : \Pi A{:}LType.\ \Pi x, y{:}A.\ \Sigma a{:}A.\ \forall p{:}A \to Prop.\ p(a) \supset p(x) \wedge p(y).$

The auxiliary entities read as follows: for any type A in $LType$ and forall $x, y{:}A$, $AND(A, (x, y))$ is a pair (a, f) such that forall $p{:}A \to Prop$, $f(p)$ is a proof that $p(a)$ implies both $p(x)$ and $p(y)$. Then, and is defined as the first projection $\pi 1$ of the auxiliary object:

(14) $and = \lambda A{:}LType.\lambda x, y, z{:}A.\ \pi_1(AND(A, x, y))$

With these in mind, let us now look at the existing proposal as regards dot-types and its proper formalization as this was provided in (Luo, 2010). The whole idea of forming a dot-type is informally based on the fact that to form a dot-type $A \bullet B$, its constituent types A and B should not share common parts/components. For example, the following two cases cannot be dot-types since they both share components:

(15) $\textsc{Phy} \bullet \textsc{Phy}$

(16) $\textsc{Phy} \bullet (\textsc{Phy} \bullet \textsc{Info})$

Definition 3.1 (components) *Let $T : Type$ be a type in the empty context. Then, $\mathcal{C}(T)$, the set of components of T, is defined as:*

$$\mathcal{C}(T) =_{df} \begin{cases} \textsc{Sup}(T) & \text{if the NF of } T \text{ is not } X \bullet Y \\ \mathcal{C}(T_1) \cup \mathcal{C}(T_2) & \text{if the NF of } T \text{ is } T_1 \bullet T_2 \end{cases}$$

where $\textsc{Sup}(T) = \{T' \mid T \leq T'\}$.

Formation Rule

$$\frac{\Gamma\ valid \quad \langle\rangle \vdash A : Type \quad \langle\rangle \vdash B : Type \quad \mathcal{C}(A) \cap \mathcal{C}(B) = \emptyset}{\Gamma \vdash A \bullet B : Type}$$

Introduction Rule

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B \quad \Gamma \vdash A \bullet B : Type}{\Gamma \vdash \langle a, b\rangle : A \bullet B}$$

Elimination Rules

$$\frac{\Gamma \vdash c : A \bullet B}{\Gamma \vdash p_1(c) : A} \qquad \frac{\Gamma \vdash c : A \bullet B}{\Gamma \vdash p_2(c) : B}$$

Computation Rules

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B \quad \Gamma \vdash A \bullet B : Type}{\Gamma \vdash p_1(\langle a, b\rangle) = a : A} \qquad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B \quad \Gamma \vdash A \bullet B : Type}{\Gamma \vdash p_2(\langle a, b\rangle) = b : B}$$

Projections as Coercions

$$\frac{\Gamma \vdash A \bullet B : Type}{\Gamma \vdash A \bullet B <_{p_1} A : Type} \qquad \frac{\Gamma \vdash A \bullet B : Type}{\Gamma \vdash A \bullet B <_{p_2} B : Type}$$

Coercion Propagation

$$\frac{\Gamma \vdash A \bullet B : Type \quad \Gamma \vdash A' \bullet B' : Type \quad \Gamma \vdash A <_{c_1} A' : Type \quad \Gamma \vdash B = B' : Type}{\Gamma \vdash A \bullet B <_{d_1[c_1]} A' \bullet B' : Type}$$

where $d_1[c_1](x) = \langle c_1(p_1(x)), p_2(x)\rangle$.

$$\frac{\Gamma \vdash A \bullet B : Type \quad \Gamma \vdash A' \bullet B' : Type \quad \Gamma \vdash A = A' : Type \quad \Gamma \vdash B <_{c_2} B' : Type}{\Gamma \vdash A \bullet B <_{d_2[c_2]} A' \bullet B' : Type}$$

where $d_2[c_2](x) = \langle p_1(x), c_2(p_2(x))\rangle$.

$$\frac{\Gamma \vdash A \bullet B : Type \quad \Gamma \vdash A' \bullet B' : Type \quad \Gamma \vdash A <_{c_1} A' : Type \quad \Gamma \vdash B <_{c_2} B' : Type}{\Gamma \vdash A \bullet B <_{d[c_1,c_2]} A' \bullet B' : Type}$$

where $d[c_1, c_2](x) = \langle c_1(p_1(x)), c_2(p_2(x))\rangle$.

Figure 1: The rules for dot-types.

The rules for the dot-types are given in Figure 1, as given in (Luo, 2012b). The notion of dot-type captures copredication in a nice way: it is both formal and suitable for MTT-semantics. The question is whether this account gives us correct individuation criteria. In order to test this, we check it against the Coq proof-assistant (The Coq team, 2007), based on the formal development as considered in (Luo, 2011). In effect, we define in Coq the dot-type PHY • INFO and define $Book$ to be the Σ-type that encodes Pustejovksy's qualia structure; as a consequence, $Book$ is a subtype of PHY • INFO. We further define *mastered* and *picked up* to be of type INFO $\rightarrow Prop$ and PHY $\rightarrow Prop$, respectively, and further provide a tactic to enhance automation, the details of which are out of the scope of this paper. Lastly, the quantifier $three$ is defined.[6]

```
Definition CN:=Set.
Parameter Man Human:CN.
Parameter John:Man.
(* Phy dot Info *)
Parameter Phy Info : CN.
Record PhyInfo:CN:=mkPhyInfo{phy:>
Phy;info:>Info}.
(*Book as Sigma-type with PhyInfo &
BookQualia*)
Parameter Hold:Phy->Info->Prop.
Parameter R:PhyInfo->Prop.
Parameter W:Human->PhyInfo->Prop.
Record BookQualia (A:PhyInfo):Set:=
  mkBookQualia {Formal:Hold A A;
                Telic:R A;
                Agent:exists
                h:Human, W h A }.
Record Book:Set:=mkBook{Arg:>
PhyInfo;Qualia:BookQualia Arg}.
Ltac AUTO:=cbv delta;intuition;try
repeat congruence;jauto;intuition.
Parameter mastered:Human->Info->Prop.
Parameter picked_up:Human->PhyProp.
Parameter AND: forall A:Type, forall
x y:A, sigT(fun a:A=>forall p:A->
Prop,p(a)->p(x) /\p(y)).
Definition and:= fun A:Type, fun x
y:A=>projT1(AND x y).
Definition Three:=fun(A:CN)(P:A->
Prop)=>exists x:A,P x/\(exists y:A,
P y/\(exists z:A,P z/\x<>y/\y<>z/\
x<>z)).
```

With these in line, let us see whether the correct predictions are being made with respect to individuation criteria. What we need to capture is the following entailment:

(17) John picked up and mastered three books ⇒ John picked up three books and John mastered three books

Basically, what we need to be able to get is a situation where three distinct informational as well as physical objects are involved. We formulate this as a theorem to be proven in Coq:

```
Theorem DT:(Three Book)(and(PhyInfo
->Prop)(picked_up John)(mastered
John))->(Three Book)(picked_up
John)/\(Three Book)(Mastered John).
```

Indeed, this can be proven in Coq. What we can further prove is the entailment that from John picked up and mastered three books, it follows that John picked up three physical objects and mastered three informational objects. In Coq notation:

```
Theorem DT:(Three Book)(and(PhyInfo
->Prop)(picked_up John)(mastered
John))->(Three Phy)( picked_up John)
/\(Three Info)(Mastered John).
```

This can be proven as well.[7]

It seems in this respect, that the account gives the correct predictions as regards individuation criteria and counting. This can be seen as an advantage compared to approaches like Asher's (2011), which gives the correct results after some additional assumptions on accommodation are made (which really complicate the account), while they further make it too permissive as to allow the following (see (Gotham, 2014)):

(18) # Fred picked up and mastered a stone.

[5] See (Chatzikyriakidis and Luo, 2012) for more details on the universe $LType$, its motivation as well as (some of) its introduction rules.

[6] $Three$ is defined as follows: forall A of type CN and given a predicate $P{:}A \rightarrow Prop$, there exist three elements, x,y and z, that are different, which are true of P.

[7] Those that wish to prove this on their own, the tactics to prove both of the examples are: $compute, intro, destruct$ AND, $case\ a\ with$ $(ThreeBook), AUTO, AUTO$.

On the other hand, the claim made by (Gotham, 2014) that the dot-type account as this is given by (Luo, 2010) cannot capture the facts, is shown to be incorrect on the basis of what we have presented here. Gotham's account predicts the correct results as well, but we believe at the expense of additional complications (e.g. the introduction of $R - compressible$ pluralities), that the present account does not introduce.

Thus, the account proposed for dot-types is not only formally sound but also gives the correct results with respect to counting and individuation criteria without the need of additional machinery. We take this to be a clear advantage over the other accounts.

4 The Case of *newspaper*: a Proposal for Linear Dot-types

Cases like *book* or *lunch*, being subtypes of dot-types, seem to have clear properties that are captured with the existing formalization given for dot-types. There is however a more problematic case, famously exemplified by the word *newspaper*, which seem to require a different, more restrictive treatment. First of all, *newspaper* is associated with three rather than two senses, i.e. institution (19), informational object (20) and physical object (21) as the examples below illustrate:

(19) The newspaper was sued on moral grounds.

(20) He read the newspaper.

(21) He picked up the newspaper.

Now, when it comes to the use of two different senses in the context of the same sentence, a number of strange restrictions appear. The physical object sense can be used along with the informational sense, in the same way as in the case of *book*, but the organizational sense (newspaper as an institution) cannot be used copredicatively with any of the other two senses (examples from (Antunes and Chaves, 2003)):

(22) # That newspaper is owned by a trust and is covered with coffee.

(23) # The newspaper fired the reporter and fell off the table.

(24) # John sued and ripped the newspaper.

Similar words with multiple senses that further involve similar restrictions are also discussed in (Retoré, 2014). There, a multi-sorted higher order logic is used[8] and every word is associated with a kind of basic type along with a number of coercions that can coerce this basic type into additional types. So in the case of book one gets the principal lambda-term $\lambda x.const(x){:}v \rightarrow t$ where v stands for event and two optional lambda-terms, $Id{:}v \rightarrow v$ and $f_a{:}v \rightarrow a$ where a stands for type *artifact*, a subtype of physical objects. The optional terms are declared as rigid, meaning that if one of the coercions is used, the other one cannot and vice versa. For the case of dot-types like *book* the optional lambda terms are dubbed as flexible, meaning that the coercions can be used simultaneously. This is indeed an interesting account. However, the exact nature of the rigid and flexible coercions are not defined formally, and it is rather unclear how such a specification can be made. Furthermore, for cases like newspaper, such an approach will not work. This is because, in the case of the coercion from $f_a{:}a \rightarrow i$ (artifact to informational object), this has to be defined as both rigid and flexible at the same time. Flexible, because we want this to be possible with the physical sense, while rigid because we want this not to be possible with the organizational sense. Furthermore, the account is based on the idea that there is always a principal lambda term. For example, in the case of *Book* the physical sense is chosen. How is this sense chosen is something that it is not explained. The question of why the physical rather than the informational aspect is chosen as the principal sense is something that is left unanswered.

The data with respect to *newspaper* get further complicated. As we have seen, the organizational aspect cannot be used with any of the other two aspects. However, this is not without exceptions. There are cases this restriction seems to disappear, allowing the organizational aspect to appear with any of the two other senses:

[8] The meta-language for the system in (Retoré, 2014) is Girard's system F rather than the simply typed λ-calculus as in Church's simple type theory (Church, 1940) as used by Montague.

(25) The newspaper you are reading is being sued by Mia.

However, if you look at the examples that allow this kind of constructions, it seems that they are of a specific kind. Most specifically all these cases involve a some kind of modification, e.g. a relative clause as in the above example, or adjectival modification as in the (22):

(26) The most provocative newspaper of the year has been sued by the government.

(27) The newspaper he just grabbed from the newsstand is doing well in the stock market.

The pattern seems to be the following: the organizational aspect cannot be used with any of the other two aspects, unless one aspect is taking part in a modified CN construction. In case, this happens the organizational aspect can be used along the other aspects. The account as proposed in (Pustejovsky, 1995) for *newspaper* cannot deal with these phenomenon and as far as we know, no formal account has been proposed for these cases. This is what we want to discuss here. The original account of dot-types in (Luo, 2010) among others will face similar problems. The dot-type $\text{INST}\bullet(\text{PHY}\bullet\text{INFO})$ will suffer the problem of predicting examples (19)-(21) to be ok contrary to fact. In what follows, we discuss a solution to this extent by proposing to treat these cases by extending the dot-type system to further include resource sensitive dot-types, i.e. linear dot-types.

Linear Dot-types: a Tentative Proposal. It is clear from what we see from the data that we are dealing with a situation where the dot-type is resource sensitive, in the sense of linear logic (Girard, 1987) or Lambek calculus (Lambek, 1958). For example, in linear logic, the rules of weakening and contraction are not available and this has a number of consequences. One of them is that one is has to use an assumption exactly once. An assumption, once used, is not re-usable anymore. It seems that this idea, is quite close to what we need for the *newspaper* case. We need an additional version of the dot-type, more specifically a linear version of the dot-type. This version will be closed related to the tenser product in linear logic and the usual dot-type, one of the important feature being that if one of its components has been used, the other one cannot be used any more.

Let us represent this linear dot-type as $A \ominus B$. We can further have combinations of regular and linear dot-types. In the case of newspaper what we need is the type $\text{INST} \ominus (\text{PHY} \bullet \text{INFO})$. With this type, we can take care of examples like (19) to (21) (these are also taken care of with a regular dot-type), while at the same time excluding examples (22-24) (that would be predicted to be ok with a regular dot-type).

Note that the examples like (25) can be accounted for without employing the linear version of dot-types. For instance, the semantics of (25) can be given as $sue(n)$ where $n : \Sigma(Newpaper, read)$ and $sue : Inst \rightarrow Prop$, because we have $\Sigma(Newpaper, read) < Newspaper < Inst \bullet (P \bullet I) < Inst$. The question of course is when do we use a linear dot-type and when a regular dot-type. In order to solve this problem, one can use local coercions, i.e. subtyping assuptions localized in terms (or judgments), as proposed in (Luo, 2010; Luo, 2012b). Local coercions have been used in (Luo, 2011) to deal with cases of homophony and in (Asher and Luo, 2012) to give semantics of linguistic coercions in sophisticated situations. Local coercions are only effective locally for some terms (expressions in type theory). They may be introduced into terms by the following rule (intuitively, the coercions declared locally are only effective in the expressions in the scope of the keyword **in**):

$$\frac{\Gamma,\ A <_c B \vdash J}{\Gamma \vdash \textbf{coercion}\ A <_c B\ \textbf{in}\ J}$$

where J is any of the following four forms of judgement:

$$k : K, \quad k = k' : K, \quad K\ kind, \quad \text{and} \quad K = K'.$$

For instance, with $J \equiv k : K$, we have

$$\frac{\Gamma,\ A <_c B \vdash k : K}{\Gamma \vdash \textbf{coercion}\ A <_c B\ \textbf{in}\ k : K}$$

In the case of *newspaper*, what we need is to consider two local coercions: $Newspaper < \text{INST} \bullet$

$(P \bullet I)$ in interpreting the cases where the ordinary dot-type should be used and $Newspaper < \text{INST} \ominus (\text{PHY} \bullet \text{INFO})$ in interpreting the cases where the linear dot-type should be used. For example, the following (28) will give a correct interpretation of (25):

(28) **coercion** $Newspaper < \text{INST} \bullet (\text{PHY} \bullet \text{INFO})$ **in** $[\![(25)]\!]$

while the following would not be accepted:

(29) # **coercion** $Newspaper < \text{INST} \ominus (\text{PHY} \bullet \text{INFO})$ **in** $[\![(22)]\!]$

We believe that this gives a satisfactory account of a problem that as far as we know has not received a treatment up to now.

However, it has to be kept in mind that we have not formally treated the linear dot-type $A \ominus B$. One of the reasons for this is that, in order to do this, we need to formally study how to incorporate coercive subtyping into a resource sensitive logical system. Put in another way, one needs to study an MTT augmented with resource sensitive contextual segments and its coercive subtyping extension. We leave this as future work.

5 Conclusion

We have discussed dot-types with respect to their counting criteria and have shown that the MTT account proposed captures the fact correctly. The proof-assistant Coq was used in order to verify that the correct inferences are predicted. The account was shown not only to produce the correct results but to do so without resorting to serious extra complications of the original account (actually none is needed). Furthermore, the case of *newspaper* was discussed and a solution based on the introduction of linear dot-types combined with local coercions was provided. The issue of introducing linear dot-types in a formal way presupposes a linear version of type theory that at the moment we do not have. Thus, we leave this issue as a subject of future research.

References

S. Antunes and R.P Chaves. On the licensing conditions of co-predication. In *Proc of the 2th Inter. Workshop on Generative Approaches to the Lexicon (GL 2007)*, 2003.

N. Asher. A type driven theory of predication with complex types. *Fundamenta Informaticae 84 (2)*, 151-183, 2012.

N. Asher. *Lexical Meaning in Context: a Web of Words*. Cambridge University Press, 2012.

N. Asher and Z. Luo. Formalisation of coercions in lexical semantics. *Sinn und Bedeutung 17, Paris*, 2012.

Mark C Baker. *Lexical categories: Verbs, nouns and adjectives*, volume 102. Cambridge University Press, 2003.

C. Bassac, B. Mery, and C. Retoré. Towards a type-theoretical account of lexical semantics. *Journal of Logic, Language and Information*, 19(2), 2010.

M.J. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, 1985.

E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, 1967.

S. Chatzikyriakidis. Adverbs in a modern type theory. In N. Asher and S. Soloviev, editors, *Proceedings of LACL2014. LNCS 8535*, pages 44–56, 2014.

S. Chatzikyriakidis and Z. Luo. An account of natural language coordination in type theory with coercive subtyping. In Y. Parmentier and D. Duchier, editors, *Proc. of Constraint Solving and Language Processing (CSLP12). LNCS 8114*, pages 31–51, Orleans, 2012.

S. Chatzikyriakidis and Z. Luo. Adjectives in a modern type-theoretical setting. In G. Morrill and J.M Nederhof, editors, *Proceedings of Formal Grammar 2013. LNCS 8036*, pages 159–174, 2013.

S. Chatzikyriakidis and Z. Luo. Natural language reasoning using proof-assistant technology: Rich typing and beyond. In *Proceedings of EACL2014*, 2014.

S. Chatzikyriakidis and Z. Luo. Natural Language Inference in Coq. *Journal of Logic, Language and Information*, 23(4):441–480, 2014.

A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5(1), 1940.

The Coq Team. *The Coq Proof Assistant Reference Manual (Version 8.1), INRIA*, 2007.

P.T. Geach. *Reference and Generality: An examination of some Medieval and Modern Theories*. Cornell University Press, 1962.

J.-Y. Girard. Linear logic. *Theoret. Comput. Sci.*, 50, 1987.

G. Gonthier. A computer checked proof of the four colour theorem, 2005.

Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell OConnor, Sidi Ould Biha, et al. A machine-checked proof of the odd order theorem. In *Interactive Theorem Proving*, pages 163–179. Springer, 2013.

M. Gotham. *Copredication, quantification and individuationy*. PhD thesis, University College London, 2014.

J. Lambek. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3), 1958.

X. Leroy. The compcert c verified compiler: Documentation and users manual. http://compcert.inria.fr/man/manual.pdf, 2013.

Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science*. Oxford University Press, 1994.

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

Z. Luo. Type-theoretical semantics with coercive subtyping. *Semantics and Linguistic Theory 20 (SALT20), Vancouver*, 2010.

Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. *Logical Aspects of Computational Linguistics (LACL'2011). LNAI 6736*, 2011.

Z. Luo. Common nouns as types. In D. Bechet and A. Dikovsky, editors, *Logical Aspects of Computational Linguistics (LACL'2012). LNCS 7351*, 2012.

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

Z. Luo. Formal Semantics in Modern Type Theories: Is It Model-theoretic, Proof-theoretic, or Both? *Invited talk at Logical Aspects of Computational Linguistics 2014 (LACL 2014), Toulouse. LNCS 8535*, pages 177–188, 2014.

Z. Luo, S. Soloviev, and T. Xue. Coercive subtyping: theory and implementation. *Information and Computation*, 223:18–42, 2012.

P. Martin-Löf. An intuitionistic theory of types: predicative part. In H.Rose and J.C.Shepherdson, editors, *Logic Colloquium'73*, 1975.

P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

R. Montague. *Formal Philosophy*. Yale University Press, 1974. Collected papers edited by R. Thomason.

B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. Oxford University Press, 1990.

J. Pustejovsky. *The Generative Lexicon*. MIT, 1995.

A. Ranta. *Type-Theoretical Grammar*. Oxford University Press, 1994.

C. Retore. The montagovian generative lexicon Tyn: a type theoretical framework for natural language semantics. In R. Matthes and A. Schubert, editors, *Proc of TYPES2013*, 2013.

G. Sundholm. Proof theory and meaning. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic III: Alternatives to Classical Logic*, pages 471–506. Reidel, 1986.

G. Sundholm. Constructive generalized quantifiers. *Synthese*, 79(1):1–12, 1989.

T. Xue and Z. Luo. Dot-types and their implementation. *Logical Aspects of Computational Linguistics (LACL 2012). LNCS 7351*, 2012.