Noname manuscript No.
(will be inserted by the editor)

Adjectival and Advebial Modification: The View from Modern Type Theories

Stergios Chatzikyriakidis · Zhaohui Luo

Received: date / Accepted: date

Abstract In this paper we present a study of adjectival/adverbial modification using Modern Type Theories (MTTs), i.e. type theories within the tradition of Martin-Löf. We present accounts in MTTs for various issues concerning adjectival/adverbial modification and argue that MTTs can be used as an adequate language for interpreting NL semantics. MTTs are not only expressive enough to deal with a range of modification phenomena but are further well-suited to perform reasoning tasks that can be easily implemented (e.g. in proof-assistants) given their proof-theoretic nature.

In MTT-semantics, common nouns are interpreted as types rather than predicates. Therefore, in order to capture the semantics of adjectives adequately, one needs to meet the challenge to model CNs modified by adjectives as types. To explicate that this can be done successfully, we first look at the mainstream classification of adjectives, i.e., intersective, subsective and non-subsective adjectives. There, we show that the rich type structure in MTTs, along with a suitable subtyping framework, offers adequate mechanisms to model these cases. In particular, this modelling naturally takes care of the characterising inferences associated with each class of adjectives. Then, more advanced issues on adjectival modification are discussed: (a) degree adjectives, (b) comparatives and (c) multidimensional adjectives. There, it is shown that the use of indexed types can be usefully applied in order to deal with these cases.

In the same vein, the issue of adverbial modification is discussed. We study two general typings for sentence and VP adverbs respectively. It is shown that the rich type structure in MTTs also provides useful organisational mechanisms in giving formal semantics. In particular, we discuss the use of Σ-types to capture the veridicality/non-veridicality distinction and further discuss cases of intensional adverbs using the type theoretic notion of context (i.e. without resorting to intensional typing). We further look at manner, subject and speech act adverbials and propose solutions using MTTs.

Finally, we show that the current proof technology can help mechanically check the associated inferences. A number of our proposals concerning adjectival and adverbial modification have been formalised in the proof assistant Coq and many of the associated inference patterns are checked to be correctly captured.

1 Introduction

The main subject of study of this paper is the modelling of adjectival and adverbial modification using semantics based on modern type theories (or MTT-semantics for short) [61, 43]. It is a well known fact

Stergios Chatzikyriakidis
Centre for Linguistic Theory and Studies in Probability, University of Gothenburg
E-mail: stergios.chatzikyriakidis@gu.se

Zhaohui Luo
Dept of Computer Science, Royal Holloway, Univ of London
Egham, Surrey TW20 0EX, U.K
E-mail: zhaohui.luo@hotmail.co.uk

that adjectival and adverbial modification is a notoriously difficult issue to tackle adequately. The main reason behind this is that the adjectival/adverbial classes are largely non-homogeneous semantic classes where a strict classification according to semantic properties is quite a difficult task. Furthermore, an additional difficulty arises in pursuing such a task: given that common nouns are interpreted as types rather than predicates in MTT-semantics, there is an additional major challenge to overcome, i.e. how to model CNs modified by various classes of adjectives as types in an adequate way. In this paper we take up these challenges.

In this introduction, we shall first summarize the issues as regards adjectival and adverbial modification that we will take up in this paper and briefly discuss how these have been dealt with in the formal semantics literature, mostly within the Montagovian tradition. We want in this respect to exemplify the complex nature of the problem as well as to provide a background on the phenomena that we will attempt to account for. Some introductory informal notes on modification from the perspective of MTTs will be given setting the context for the proper introduction to MTT semantics in chapter 2 as well as to the analysis that follows in chapters 3 and 4, 5 and 6.

Adjectival Modification. Starting off with adjectives, a first coarse-grained distinction originating in earlier approaches within the Montagovian tradition [52, 29, 58, 59], recognizes three main categories of adjectives: intersective, subsective and non-subsective (with two subcategories) along with their respective associated inferences they give rise to:

(1) Intersective: $Adj(N)(x) \Rightarrow Adj(x) \wedge N(x)$

(2) Subsective: $Adj(N)(x) \Rightarrow N(x)$

(3) Non-subsective (privative): $Adj(N)(x) \Rightarrow \neg N(x)$

(4) Non-subsective (non-committal): $Adj(N)(x) \Rightarrow ?$

An example of the above adjective will be *black*. A black man for example is someone who is both black and a man. In a sense the blackness is not contingent on being a man. Thus, a black man is also a black human, a black animal and so on. To the contrary, subsective adjectives are contingent to the noun class they modify. Thus, a skilful surgeon is someone who is skilful as a surgeon, but we do not know if he is a skilful in general. Non-subsective adjectives on the other hand, involve two distinct categories: privative adjectives where the adjective-noun property entails the negation of the property of being a noun. *Fake* is a prototypical privative adjective. Lastly, non-commital adjectives involve adjectives that do not commit us to any of the aforementioned inferences. An example of such an adjective is *alleged*. An alleged thief might or might not be a thief.

All adjectives receive a unified intensional type from properties to properties, i.e. $(s \to (e \to t)) \to (s \to (e \to t))$,[1] in Montagovian treatments [29, 58]. Such a type can deal with intensional adjectives like *former* and *alleged* among others. This is indeed needed in order to avoid unwanted inferences like the one shown below:

(5) John is an alleged thief $\Rightarrow$ John is a thief

Intensional typing does the trick for cases like the one above. The main idea here is to generalize to the worst case, a technique frequently used in Montague semantics (the treatment of quantifiers being such a case). However, in order to take care of the inferences associated with the different types of adjectives, meaning postulates are introduced, which specify the exact semantics for each subclass. Such a postulate, the one for subsective adjectives, is shown below (modified from [58] via ignoring intensional operators):

(6) $\forall Q : e \to t. \forall x : e.\ ADJ(Q, x) \supset Q(x)$

Generalizing to the worst case, in the case of adjectives the use of intensional typing across the board, i.e also for adjectives (as well as adverbs) that are not intensional is counterintuitive. Furthermore, the use of meaning postulates will not be needed at least for intersective and subsective adjectives, if one

[1] Within the simple type theory used in Montague Grammar, e is the type of individuals, t is the type of truth-values and s the type of world-time pairs.

moves to a rich type theory with subtyping, like the one we use in this paper. This has been already exemplified in [7] and it is going to be discussed in this paper as well.

Now, anyone who has looked at the semantics of adjectives in more detail knows that the classification into intersective, subsective and non-subsective adjectives, even though a useful one, is rather coarse-grained for a number of cases. In order to have a deeper understanding of the semantics of adjectives (or aspects of their semantics to be more precise), a number of further issues that require a more fine-grained semantic treatment need be taken care of. One issue at hand involves *gradable adjectives*. Gradable adjectives have been traditionally treated as involving an extra parameter, that of a DEGREE variable that can be bound by different morphosyntactic forms of the adjective, i.e. its comparative [70, 25] among others. One standard way of looking at this extra complication, is that gradable adjectives involve some kind of measurement. Usually, this measurement is taken to be a degree argument, which is then taken to be the main difference between gradable and non-gradable adjectives: the presence of a degree argument. This argument has been proposed to be formally encoded in the typing as in[3, 70, 26], or not as in [35, 50, 32, 69]. To give an example, the adjective small, will involve the following definition according to the first view:

(7) $[\![small]\!] = \lambda d{:}Degree.\lambda x{:}e.Height(x) \leq d$

Another issue concerning adjectives is the way that the positive form is connected with the comparative and the rest of its forms.

Lastly, there are a number of adjectives, the so-called *multidimensional adjectives* that present added complications; their meanings do not just involve one dimension (e.g., the dimension of height in the case of *tall*), but more than one. Classical cases involve adjectives like *healthy* and *sick* or even adjectives like *big*. The intuition is that adjectives like e.g. *healthy* quantify over a number of dimensions, e.g. blood pressure, cholesterol etc. [63]. Similarly, *big* may involve different dimensions like *height*, *width* etc. Cases of multidimensional adjectives rely heavily on the context with respect to the dimensions that need to be satisfied in order for an utterance to be felicitous. The insight provided by Sassoon is that there are two classes of multidimensional adjectives are distinguished, [63], positive and negative. Positive adjectives incude cases like *healthy*, while negative ones cases like *sick*. The difference in each case concerns the form of quantification over dimensions in each case, universal quantification over dimensions for positive multidimensional adjectives, existential quantification over dimensions for negative ones. Consider the following examples:

(8) Dan is healthy except with respect to blood pressure

(9) # Dan is sick except with respect to blood pressure

Examples like above exemplify the latter claim. In this sense, the two classes of adjectives have also different inferential properties. A sentence like John is healthy implies that John is healthy with respect to all "health" dimensions while for *sick*, what we get is an entailment that John is not healthy across some dimension.

Adverbial Modification. Adverbials, similarly to adjectives, are a largely non-homogeneous class. Given this non-homogeneity, it is not surprising that various classifications having been proposed throughout the years. According to one of the most prominent ones, that of Ernst and Maienborn [21, 46], adverbs receive a tripartite classification with further subclassifications for each class: a) predicational, b) participant oriented and c) functional adverbials.[2]

Predicational adverbs comprise the main bulk of adverbs, its main subcategories including sentence and verb related adverbs. Sentence adverbs are further classified into subject oriented adverbs like *arrogantly*, speaker oriented adverbs that include speech act adverbials like *honestly*, epistemic adverbials like *possibly* and domain adverbs like *botanically*, while verb related adverbials into mental attitude adverbs

[2] Maienborn [46] uses only semantic criteria, while Ernst [21] mostly concentrates on the syntax of adverbs. See the individual pieces of work for more details.

like *reluctantly*, manner adverbs like *skillfully* and degree adverbs like *deeply*. *Participant oriented adverbials* on the other hand include adverbials that introduce a new entity/entities to the situation/event described by the proposition in question. Examples of this type of adverbials include cases like *with a knife*, *with a gun* etc. Lastly, *functional adverbials* include adverbials where some kind of quantification is involved like *usually, never* etc.

The main bulk of proposals as regards adverbials falls within two lines of approach: a) the operator approach proposed within the Montagovian tradition and b) the (neo)-Davidsonian event-related approach. Both ways of looking at the issue have their merits as well as their disadvantages. The operator approach (see [51, 67, 29] among others), views adverbs as functors which return the same type as that of their argument. Furthermore, the operator approach distinguishes between adverbs that take a truth value (or a proposition) to return a truth value (or proposition) and adverbs that take a set (or concept) to return a set (or concept):

(10) Extensional: $(e \to t) \to (e \to t)$
Intensional: $(s \to (e \to t)) \to (s \to (e \to t))$

(11) Extensional: $t \to t$
Intensional: $(s \to t) \to (s \to t)$

The above typings correspond to sentence adverbs (e.g. evaluative adverbs like *fortunately*) and VP adverbs (e.g. manner adverbs). Their intensionalized typing versions make the rather welcoming prediction that in VP-adverbs, opaque contexts should arise for the object but not for the subject, a prediction which is borne out from the facts (see [29, 67] and [46] for a summary). To give an example: imagine a VP-adverb like *intentionally* in a sentence like *Oedipus intentionally married Jocaste*. Under the intensionalized VP-adverb typing, we have that *Oedipus intentionally married his mother* does not follow. However, it does follow that the son of Laius intentionally married Jocaste. Similarly, it correctly predicts that for sentence intensional adverbs, opacity should arise for both the subject and the object. Thus, in a sentence like Oedipus allegedly married Jocaste, both *the son of Laius allegedly married Jocaste* and *Oedipus allegedly married his mother* do not follow. On the other hand, approaches within the tradition initiated by Davidson [17], argue that adverbs can be seen as providing restrictions w.r.t. the event denoted by the sentence/proposition in each case. In effect, adverbs in these approaches are assumed to modify the event in some way. For example a manner adverb like *beautifully* is taken to involve event modification. This is general embedded within a theory which takes the semantics of semantics to boil down to a conjunction of predicates applied to either entity or event arguments. Consider the sentence below:

(12) Mary dances beautifully

The classic analysis manner adverbs within the neodavidsonian tradition is one where manner adverbs are seen as modifiers over events. Informally the sentence means that there is a dancing event, of which Mary is the agent, and this event is a beautiful one. Somewhat more formally:

(13) $\exists e[dancing(Mary, e) \wedge beautiful(e)])$

The problem here, as already noted in the literature, is that the manner of the event and not the event itself is modified. Thus, one might want to introduce *manner* in the ontology of types/basic predicates (depending on whether we have a typed or a non-typed system). We believe that this extension can be done very easily within a rich type system as the one we are endorsing in this paper.

Other types of adverbials that we are going to be looking in this paper, include subject oriented and and agent oriented adverbs. In the former case, we find adjectives like stupidly, where in this case not only a manner of the event is denoted but also the agent of the event:

(14) John stupidly opened the door $\Rightarrow$ the manner that John opened the door was stupid

Agent oriented adverbs on the other hand, seem to provide commentary with respect to the utterance. An adverb like *frankly* seems to imply that a sentence of the form *frankly* P means something like "I *frankly* tell you that P". We will try to show, as in the case of manner adverbs, that the elaborate typing mechanisms of MTTs can be used in order to get this fine-grained meaning nuances associated with the different kinds of adverbs.

Adjectival/Adverbial Modifications in MTT-semantics: Some preliminary notes The major aim of the paper is to study how to deal with adjectival and adverbial modification in MTT-semantics. As briefly mentioned above, in order to model adjectival modification adequately in MTT-semantics, one has to meet the challenge of how to interpret CNs modified by various classes of adjectives as *types* rather than predicates.

As we know, even in MTTs which have many inductively defined types, there are less types than predicates/sets. To put it in another way, intuitively, types are *manageable* sets. For example, whether an object a is of type A, notation $a : A$, is decidable (i.e., type checking is decidable, in a technical jargon), while whether an element s is in a set S, notation $s \in S$, is not. This gives an advantage of types over sets in the following sense: given that MTTs are proof-theoretically specified, they can be implemented effectively in computers – systems called proof assistants. When a formal semantics is given in MTTs, it can be directly implemented in a proof assistant which, among other things, provides tools for inference based on the implemented MTT-semantics.

In this paper, we shall show that various type constructors in MTTs provide us with adequate mechanisms to model adjectival/adverbial modification of various kinds. For instance, we will show how to employ Σ-types for intersective adjectives, disjoint union types for privative adjectives, and polymorphic Π-types with universes and Σ types for subsective adjectives.[3]} In general, we shall illustrate that, using the rich type structure of MTTs, we can provide an account of a number of issues in adjectival/adverbial modification.

In §2, we introduce the core features of modern type theories, emphasizing those relevant to this paper, setting up the background knowledge and notation. In §3 and §4 we deal with a number of aspects of adjectival modification: a) the traditional classification into intersective, subsective and non-subsective adjectives in §3, and b) gradable and multidimensional adjectives in §4. In §5, we look at adverbial modification. There, building on work by [6], we show how the rich typing constructs of MTTs can give us a way out with respect to veridicality, intensional adverbs and some aspects concerning manner and X-oriented adverbs. Lastly, in §6, we check some of the proposals in this paper in terms of their inferential properties. We show how MTTs can be used in this respect both from a theoretical as well as an implementational point of view. With respect to the latter, we implement some of the ideas in this paper in the Coq proof assistant [16], showing its potential use to the study of NL inference.

Remark 1 It may be useful to emphasise that the scope of the current paper is rather moderate. Our aim, as already mentioned, is to show the way modification can be treated in a framework where CNs are modelled as types instead of predicates (or funcational sets of type $e \to t$). As such, this paper should not be taken as a paper that tries to compete with the other state-of-the-art approaches found in the considerably rich formal semantics literature in all the aspects that this paper discusses. This is a paper that aspires to show the way to use an alternative formal semantics theory for the study of linguistic semantics. On the other hand, we do not want this paper to be seen as framework gymnastics, i.e. a plain exercise in formal semantics using just another framework. MTTs have a number of advantages compared to simple type theories when it comes to their respective computational properties as well as their fitness to support proof-theoretic reasoning (cf., decidability and practical inference in proof assistants, as mentioned above). These two latter properties of MTT-semantics and their potential application in the study of natural language inference not only from a semantic but from a computational point of view as well, make, we believe, the ideas put forth in this paper worth pursuing.

2 MTTs with Coercive Subtyping: Introduction

In this section, we give a brief introduction to the formal semantics based on Modern Type Theories (MTTs) [61, 39, 43]. We will try and introduce MTTs by exemplifying its various features as related to

[3] See next chapter for an the explanation of these types.

linguistic semantics. A modern type theory is a variant of a class of type theories as studied by Martin-Löf [47, 48] and others, which have dependent types and inductive types, among others. Among MTTs, we are going to employ the Unified Theory of dependent Types (UTT) [36] with the addition of the coercive subtyping mechanism (see, for example, [38, 45] and below).

2.1 Type many-sortedness and CNs as types

A difference between MTT-semantics and Montague semantics lies in the interpretation of common nouns (CNs). In Montague semantics [53], the underlying logic (Church's simple type theory [14]) can be seen as 'single-sorted' in the sense that there is only one type e of all entities. The other types such as t of truth values and the function types generated from e and t do not stand for types of entities. In this respect, there are no fine-grained distinctions between the elements of type e and as such all individuals are interpreted using the same type. For example, *John* and *Mary* have the same type in simple type theories, the type e of individuals. An MTT, on the other hand, can be regarded as a 'many-sorted' logical system in that it contains many types and. In this respect, in an MTT-semantics one can make fine-grained distinctions between individuals and use those different types to interpret subclasses of individuals. For example, we can have $John : [\![man]\!]$ and $Mary : [\![woman]\!]$, where $[\![man]\!]$ and $[\![woman]\!]$ are different types.

An important trait of MTT-semantics is the interpretation of CNs as *types* [61] rather than sets or predicates (i.e., objects of type $e \to t$) as in Montague semantics. The CNs man, human, table and book are interpreted as types $[\![man]\!]$, $[\![human]\!]$, $[\![table]\!]$ and $[\![book]\!]$, respectively. Then, individuals are interpreted as being of one of the types used to interpret CNs.

This many-sortedness (i.e., the fact that there are many basic types in an MTT, but also typing constructors like Σ, Π) has the welcoming result that a number of semantically infelicitous sentences like e.g. *the ham sandwich walks*, which are however syntactically well-formed, can be explained easily given that a verb like walks will be specified as being of type $Animal \to Prop$ while the type for *ham sandwich* will be $[\![food]\!]$ or $[\![sandwich]\!]$, which is not compatible with the typing for *walks*:[4]

(15) $the\ ham\ sandwich : [\![food]\!]$

(16) $walk : [\![human]\!] \to Prop$

The idea of common nouns being interpreted as types rather than predicates has been argued in [42] on philosophical grounds as well. There, the second author argues that Geach's observation that common nouns, in contrast to other linguistic categories, have criteria of identity that enable common nouns to be compared, counted or quantified, has an interesting link with the constructive notion of set/type: in constructive mathematics, sets (types) are not constructed only by specifying their objects but they additionally involve an equality relation. The argument is then that the interpretation of CNs as types in MTTs is explained and justified to a certain extent.[5]

Interpreting CNs as types rather than predicates has also a significant methodological implication: this is compatible with various subtyping relations one may consider in formal semantics. For instance, in modelling some linguistic phenomena semantically, one may introduce various subtyping relations by postulating a collection of subtypes (physical objects, informational objects, eventualities, etc.) of the type of entities [1]. It has become clear that, if CNs are interpreted as predicates as in the traditional Montagovian setting, introducing such subtyping relations would cause problems: even some basic semantic interpretations would go wrong and it is very difficult to deal with some linguistic phenomena such as copredication satisfactorily. Instead, if CNs are interpreted as types, as in MTTs, copredication can be given a straightforward and satisfactory treatment [39].

Σ-types, Π-types, indexed types and Universes We shall introduce several dependent types and the notion of type universe.

[4] This is of course based on the assumption that the definite NP is of a lower type and not a Generalized Quantifier.

[5] See [42] for more details on this.

Dependent Σ-types. One of the basic features of MTTs is the use of Dependent Types. A dependent type is a family of types that depend on some values. The constructor/operator Σ is a generalization of the Cartesian product of two sets that allows the second set to depend on values of the first. For instance, if $[\![human]\!]$ is a type and $male : [\![human]\!] \rightarrow Prop$, then the Σ-type $\Sigma h : [\![human]\!] \,.\, male(h)$ is intuitively the type of humans who are male.

More formally, if A is a type and B is an A-indexed family of types, then $\Sigma(A, B)$, or sometimes written as $\Sigma x{:}A.B(x)$, is a type, consisting of pairs (a, b) such that a is of type A and b is of type $B(a)$. When $B(x)$ is a constant type (i.e., always the same type no matter what x is), the Σ-type degenerates into product type $A \times B$ of non-dependent pairs. Σ-types (and product types) are associated projection operations π_1 and π_2 so that $\pi_1(a, b) = a$ and $\pi_2(a, b) = b$, for every (a, b) of type $\Sigma(A, B)$ or $A \times B$.

The linguistic relevance of Σ-types can be directly appreciated once we understand that in its dependent case, Σ-types can be used to interpret linguistic phenomena of central importance, like adjectival modification (see above for interpretation of modified CNs) [61]. For example, *handsome man* is interpreted as Σ-type (17), the type of handsome men (or more precisely, of those men together with proofs that they are handsome):

(17) $\Sigma m : [\![man]\!] \,.\, [\![handsome]\!](m)$

where $[\![handsome]\!](m)$ is a family of propositions/types that depends on the man m.

Dependent Π-types. The other basic constructor for dependent types is Π. Π-types can be seen as a generalization of the normal function space where the second type is a family of types that might be dependent on the values of the first. A Π-type degenerates to the function type $A \rightarrow B$ in the non-dependent case. In more detail, when A is a type and P is a predicate over A, $\Pi x{:}A.P(x)$ is the dependent function type that, in the embedded logic, stands for the universally quantified proposition $\forall x{:}A.P(x)$. For example, the following sentence (18) is interpreted as (32):

(18) Every man walks.

(19) $\Pi x : [\![man]\!] \,.\, [\![walk]\!](x)$

Π-types are very useful in formulating the typings for a number of linguistic categories like VP adverbs or quantifiers. The idea is that adverbs and quantifiers range over the universe of (the interpretations of) CNs and as such we need a way to represent this fact. In this case, Π-types can be used, universally quantifying over the universe $\textsc{cn}$. (25) the type for VP adverbs[6] while (21) is the type for quantifiers:

(20) $\Pi A : \textsc{cn}.\ (A \rightarrow Prop) \rightarrow (A \rightarrow Prop)$

(21) $\Pi A : \textsc{cn}.\ (A \rightarrow Prop) \rightarrow Prop$

Further explanations of the above types are given after we have introduced the concept of type universe below.

Indexed types. Indexed types are special kinds of dependent types where the type depends on an index, in effect we are dealing with families of types that are indexed by a type parameter. The type parameter is usually a simple one in most cases. Examples of indexes include, for instance, the type N of natural numbers, the type $Human$ of human beings, and the type $Height$ of heights (see below). We can think for example that for $h : Human$, there is a family of types $Evt(h)$ of events that are conducted by h. In the same sense, one can also think that for $h{:}Height$, there is a family of types $Human(h)$ of humans with a height parameter. This types will be of great importance in our discussion of gradable adjectives in chapter 4.

Type Universes. An advanced feature of MTTs, which will be shown to be very relevant in interpreting NL semantics in general as well as adjectival modification specifically, is that of universes. Informally,

[6] This was proposed for the first time in [41].

$$\frac{}{Human : \textsc{cn}_H} \qquad \frac{A < Human}{A{:}\textsc{cn}_H} \qquad \frac{A{:}\textsc{cn}_H}{A{:}\textsc{cn}}$$

Fig. 1 Introduction rules for $\textsc{cn}_H$.

a universe is a collection of (the names of) types put into a type [48].[7] For example, one may want to collect all the names of the types that interpret common nouns into a universe $\textsc{cn} : Type$. The idea is that for each type A that interprets a common noun, there is a name $\overline{A}$ in $\textsc{cn}$. For example,

$$\overline{[\![man]\!]} : \textsc{cn} \quad \text{and} \quad T_{\textsc{cn}}(\overline{[\![man]\!]}) = [\![man]\!]\,.$$

In practice, we do not distinguish a type in $\textsc{cn}$ and its name by omitting the overlines and the operator $T_{\textsc{cn}}$ by simply writing, for instance, $[\![man]\!] : \textsc{cn}$. Thus, the universe includes the collection of the names that interpret common nouns. For example, in $\textsc{cn}$, we shall find the following types:

(22) $[\![man]\!]$, $[\![woman]\!]$, $[\![book]\!]$, ...
(23) $\Sigma m : [\![man]\!] \,.\, [\![handsome]\!](m)$
(24) $G_R + G_F$

where the Σ-type in (23) is the proposed interpretation of 'handsome man' and the disjoint union type in (24) is that of 'gun' (the disjoint union of real guns and fake guns – see the discussion in §3). We can furthermore use partitions of the universe $\textsc{cn}$that would correspondo to more restrictive universes that might be needed by different predicates. For example, one might assume predicates having a restriction which only allows arguments that are of type $Human$ or any of its subtypes. In that case, one can introduce the subuniverse $\textsc{cn}_H$that consists of type $Human$ and all its subtypes:

It goes without saying that the universe $\textsc{cn}$ is an open universe, where additional types can be always added.

Quantifiers and adverb typing Having introduced the universe $\textsc{cn}$, it is now possible to explain (25) and (21). The difference between this type and the type of GQs in Montague Grammar is that in the latter case we have a relation between two sets, i.e. two predicates of type $e \to t$. The type for quantifiers is shown below (ignoring intensions):

(25) $(e \to t) \to (e \to t) \to t$

The first predicate is the type for the noun, given that nouns are considered to be predicates in MG, and the second predicate is the one denoted by the verb. In MTT semantics where common nouns are types and not predicates, the relation is now between a type and a predicate. Typing has to further be polymorphic given that we have a multitude of basic types, and in principle we would want to have quantification with all these types. The solution is to have a polymorphic type extending over the universe $\textsc{cn}$, i.e. the type in (25). Thus, we first need an argument of type A, $A{:}\textsc{cn}$. This corresponds to the $\textsc{cn}$argument. As soon as we get this argument, what we get back is the type $(A \to Prop) \to Prop$. Type polymorphy will predict that the type returned will be dependent on the value of A. If $A = Man$ then the type returned will be $(Man \to Prop) \to Prop$, if $A = Human$, $(Human \to Prop) \to Prop$ and so on. For example, *some human* is of type $([\![human]\!] \to Prop) \to Prop$ given that the A here is $[\![human]\!] : \textsc{cn}$ (A becomes the type $[\![human]\!]$ in $([\![human]\!] \to Prop) \to Prop$). Then, given a predicate like $walk : [\![human]\!] \to Prop$, we can apply *some human* to get $[\![some\ human]\!]([\![walk]\!]) : Prop$. The reader can now realize how the adverb typing in (25) is to be understood.

[7] There is quite a long discussion on how these universes should be like. In particular, the debate is largely concentrated on whether a universe should be predicative or impredicative. A strongly impredicative universe U of all types (with $U : U$ and Π-types) is shown to be paradoxical [23] and as such logically inconsistent. The theory UTT we use here has only one impredicative universe $Prop$ (representing the world of logical formulas) together with an infinitely many predicative universes which as such avoids Girard's paradox (see [36] for more details).

Subtyping in Formal Semantics As briefly explained above, given the many-sortedness of MTTs, CNs can be interpreted as types. For instance, in a Montagovian setting, all of the verbs below are given the same type ($e \to t$), but in an MTT, we can have:

(26) $drive : [\![human]\!] \to Prop$
(27) $eat : [\![animal]\!] \to Prop$
(28) $disappear : [\![object]\!] \to Prop$

which have different domain types. This has the advantage of disallowing interpretations of infelicitous examples like *the ham sandwich walks*.

However, interpreting CNs by means of different types could lead to serious undergeneralizations without a subtyping mechanism. Thus, *subtyping* is crucial for MTT-semantics. For instance, consider the interpretation of the sentence 'A man talks' in Figure 2: for m of type $[\![man]\!]$ and $[\![talk]\!]$ of type $[\![human]\!] \to Prop$, the function application $[\![talk]\!](m)$ is only well-typed because we assume $[\![man]\!]$ be a subtype of $[\![human]\!]$.

Coercive subtyping [38, 45] provides an adequate framework to be employed for MTT-based formal semantics [39, 43].[8] It can be seen as an abbreviation mechanism: A is a (proper) subtype of B ($A < B$) if there is a unique implicit coercion c from type A to type B and, if so, an object a of type A can be used in any context $\mathfrak{C}_B[\_]$ that expects an object of type B: $\mathfrak{C}_B[a]$ is legal (well-typed) and equal to $\mathfrak{C}_B[c(a)]$.

As an example, in the case that both $[\![man]\!]$ and $[\![human]\!]$ are base types, one may introduce the following as a basic subtyping relation:

(29) $[\![man]\!] \ < \ [\![human]\!]$

In case that $[\![man]\!]$ is defined as a composite Σ-type (see below for details), where $male : [\![human]\!] \to Prop$:

(30) $[\![man]\!] = \Sigma h : [\![human]\!] \,.\, male(h)$

we have that (29) is the case because the above Σ-type is a subtype of $[\![human]\!]$ via the first projection π_1:

(31) $\Sigma h : [\![human]\!] \,.\, male(h) <_{\pi_1} \ [\![human]\!]$

Equipped with this coercive subtyping mechanism, the undergeneration problems can be straightforwardly solved while still retaining the ability to rule out semantically infelicitous cases like *the ham sandwich walks*. In effect, many-sortedness in MTTs turns out to be superior than single sortedness in simple type theory (at least in this respect). Furthermore, many inferences concerning the monotonicity on the first argument of generalized quantifiers can be directly captured using the subtyping mechanism. In effect an inference of the sort exemplified in the example (32) below, can be captured given that $[\![man]\!] \ < \ [\![human]\!]$:

(32) Some man runs $\Rightarrow$ Some human runs

Thus, an $x : [\![man]\!]$ can be used as an $x : [\![human]\!]$, and as such the inference goes through for 'free' in a way.

Another important trait of the coercive subtyping mechanism, which will be very important in our discussion of intersective adjectives, is that subtyping also propagates through the constructors. For example, if we have $Man < Human$ and the Σ types $\Sigma m : [\![man]\!] \,.\, [\![handsome]\!](m)$ and $\Sigma m : [\![human]\!] \,.\, [\![handsome]\!](m)$, then $\Sigma m : [\![man]\!] \,.\, [\![handsome]\!](m) < \Sigma m : [\![human]\!] \,.\, [\![handsome]\!](m)$. For more information on subtyping propagation see **ZHAOHUI ADD MOST SUITED REFERENCE**

[8] It is worth mentioning that subsumptive subtyping, the traditional notion of subtyping that adopts the subsumption rule (if $A \leq B$, then every object of type A is also of type B), is inadequate for MTTs in the sense that it would destroy some important properties of MTTs (see, for example, §4 of [45] for details).

Coercion Contexts and Local Coercions It is a well-known fact that word meanings heavily rely on contexts. In this sense, a lot of the times we need to deal with cases like the following classic meaning transfer example from [57]:

Example 1 (reference transfer) *Consider the following utterance (cf., [57]):*

(33) The ham sandwich shouts.

Assuming that the act of shouting requires that the argument be human, it is obvious that sentence (33) is not well-formed, unless it is uttered by somebody in some special extralinguistic context (e.g., by a waiter in a café to refer to a person who has ordered a ham sandwich). □

This kind of local coercions or local subtypes can be dealt with in an MTT via using the type theoretical notion of context. A context is of the following form:

$$x_1 : A_1,\ ...,\ x_n : A_n$$

where A_i is can be a data type, in which case x_i is assumed to be an object of that type, or a logical proposition, in which case the proposition A_i is assumed to be true and x_i a proof of A_i. For example, one may have the following context:

$$m : [\![man]\!],\ hproof : [\![handsome]\!](m)$$

basically an assumption that m is of type man (m is a man) and a proof that this m is handsome. The context can furthermore be extended to involve subtyping declarations of the following form:

(34) ..., $[\![ham\ sandwich]\!] < [\![human]\!]$, ...

where what we have is that type *ham* is coerced into *human* in this particular setting. The range of coercions we can perform can get more-fine grained as it has been exemplified for example in the work of [2]. The interested reader is referred there for more information on this issue. People interested in seeing the use of contexts for NL semantics, please see [61, 5] as well as [12] for the similar notion of signature.

Proof-theoretic Facet of MTT-semantics. One of the key features of formal semantics in MTTs is its proof-theoretic facet. It has been pointed out in [44] that MTT-semantics is both model-theoretic and proof-theoretic. Without getting into the details, we emphasise here the proof-theoretic characteristics of MTT-semantics: it allows us to understand the MTT-semantics in a proof-theoretic way (as logics can be understood proof-theoretically [28]) and, furthermore, allows a direct application of type theory based proof assistants such as Coq [16] in conducting practical inferences based on the MTT-semantics. This has amounted to a computational treatment of formal semantics and opens up a new avenue in semantics-based reasoning in natural language (see [10] and §6). **(ZHAOHUI: WE CAN POTENTIALLY EXPAND A LITTLE BIT HERE SINCE WE ARE ASKED TO. POSSIBLY CLARIFY WHAT MODEL THEORETIC MEANS HERE CAUSE PEOPLE THINK OF MODELS STRAIGHT AWAY WHEN THEY HEAR THAT. THIS IS WHAT THE REVIEWER DID AS WELL.)**

MTT semantics interpreting natural language semantics: Preliminaries Recapitulating somehow from the previous discussion, we present the way basic linguistic categories are interpreted in MTT-semantics:

- A sentence (S) is interpreted as a proposition of type *Prop*.
- A common noun (CN) can be interpreted as a type.
- A verb (IV) can be interpreted as a predicate over the type D that interprets the domain of the verb (ie, a function of type $D \to Prop$).
- An adjective (ADJ) can be interpreted as a predicate over the type that interprets
- A VP adverb can be interpreted as a function from predicates ($A \to Prop$) to predicates ($A \to Prop$) where the A extends over the universe CN

	Example	Montague semantics	MTT-semantics
CN	man, human	$[\![man]\!],\ [\![human]\!] : e \to t$	$[\![man]\!],\ [\![human]\!] : Type$
IV	talk	$[\![talk]\!] : e \to t$	$[\![talk]\!] : [\![human]\!] \to Prop$
ADJ	handsome	$[\![handsome]\!] : (e \to t) \to (e \to t)$	$[\![handsome]\!] : [\![man]\!] \to Prop$
VP$_{adv}$	quickly	$[\![quickly]\!] : (e \to t) \to (e \to t)$	$[\![quickly]\!] : \Pi A{:}\mathrm{CN}.(A \to Prop) \to (A \to Prop)$
Quantifier	some	$[\![some]\!] : (e \to t) \to (e \to t) \to t$	$[\![some]\!] : \Pi A{:}\mathrm{CN}.(A \to Prop) \to Prop$
MCN	handsome man	$[\![handsome]\!]([\![man]\!]) : e \to t$	$\Sigma m : [\![man]\!] \,.\, [\![handsome]\!](m) : Type$
S	A man talks	$\exists m : e.\ [\![man]\!](m) \& [\![talk]\!](m) : t$	$\exists m : [\![man]\!] \,.\, [\![talk]\!](m) : Prop$

Fig. 2 Examples in formal semantics.

- A quantifier is interpreted as a function from a type $A{:}\mathrm{CN}$ extending over the universe CN to a function from predicates over A to propositions.[9] the domain of the adjective (ie, a function of type $D \to Prop$).
- Modified common nouns (MCNs) can be interpreted by means of Σ-types (see below).

Basic examples are shown in Figure 2, along with a comparison with their counterparts in Montague semantics.

Note that this table shows some basic comparisons to Montague semantics as well as potential ways to interpret these categories. For some of these cases, other ways of interpretation will be pursued. For example, besides the use of a basic predicate type for adjectives, type polymorphy will be used for e.g. cases of subsective adjectives.

3 Intersective, Subsective and Non-subsective Adjectives

In this section, we look at the traditional formal semantics classification of adjectives [29, 30, 58] and discuss the solutions using MTTs based on earlier work of ours [7]. Historically, Ranta [61] first proposed to use Σ-types to represent adjectival modification.[10] However, the proposal was not completely working because there was no proper subtyping mechanism that is essential for Σ-types to be employed for adjectival modification.[11] This problem was solved in [39], where the second author proposed to employ coercive subtyping [38, 45]. As a consequence, CNs modified by intersective adjectives can be properly represented by means of Σ-types.

Subsective adjectival modification was then studied in [7] where the authors proposed to use the universe CN of common nouns and polymorphism in representations of subjective adjectives like *small,large*. Non-subsective adjectives have also been studied: for privative adjectives like *fake*, [40, 7] propose to use disjoint union types while, for non-committal adjectives like *alleged*, belief contexts [7]. All these proposed solutions maintain a lower type for adjectives. The rich typing provided by MTTs in conjunction with coercive subtyping [38, 45] can give us an attractive solution to all adjectives under the traditional classification into intersective, subsective and non-subsective adjectives.

3.1 Intersective and Subsective Adjectives

Intersective and subsective adjectives can be treated properly using Σ-types. Using Σ-types to represent adjectival modification, as already mentioned, was originally proposed by Ranta [61]. However, in Ranta's account, neither of the two classes can be captured successfully: for intersective adjectives, we lack a proper subtyping mechanism and, for subsective adjectives, we need to use the universe of common nouns and polymorphism.

[9] For a constructive version of generalized quantifiers see [66]

[10] Ranta [61] did not consider different classes of adjectives and we think that he mainly had intersective adjectives in his mind when considering this.

[11] It may be worth remarking that the notion of 'subset' [56] as discussed in §3.3 of [61] has been abandoned since it cannot be properly incorporated into MTTs such as Martin-Löf's type theory.

The Σ-type treatment is straightforward. In MTT-semantics, CNs are interpreted as types and adjectives as predicates, i.e. functions of type $A \to Prop$, for some type $A : CN$. Thus, an adjective like *black* (on the assumption that *black* is intersective) will be of type $[\![object]\!] \to Prop$, an adjective like *handsome* of type $[\![human]\!] \to Prop$, and so on: note that the typing for *handsome* is $[\![human]\!] \to Prop$, while our example involves the type *man*. The reason this example is well-formed is due to the subtyping relation $man < human$ and the fact that subtyping propagates through the constructors.

(35) $[\![black]\!] : [\![object]\!] \to Prop$

(36) $[\![handsome]\!] : [\![human]\!] \to Prop$

A modified CN like *handsome man* is interpreted as the following Σ-type:

(37) $\Sigma m{:}\, [\![man]\!] \,.\, [\![handsome]\!](m)$

Note that the typing for *handsome* is $[\![human]\!] \to Prop$, while our example involves the type *man*. The reason this example is well-formed is due to the subtyping relation $[\![man]\!] < [\![human]\!]$ and the fact that subtyping relations propagate through the type constructors (in our example, Σ, [39])

(38) $\Sigma m{:}\, [\![man]\!] \,.\, [\![handsome]\!](m) < \Sigma m{:}\, [\![human]\!] \,.\, [\![handsome]\!](m)$

Let us see what this account predicts as regards inference. Intersective adjectives are associated with two types of inference. The first one is shown below:**(ZHAOHUI: THE REVIEWER ASKS THAT WE HAVE BOTH THE INFORMAL AND THE FORMAL INFERENCE IN THE MAIN TEXT, IN THE SAME SENSE WE HAVE IN THE FOOTNOTE. CAN YOU DO THAT?)**[12]

(39) From Adj(N)(x) *infer* N(x).

This inference is very easily taken care of, given that the first projection of the Σ is always a coercion, in effect for any Σ type of the form $\Sigma(A, B)$, we can always infer A. Subtyping does the job here. The second inference associated with intersective adjectives is a little bit trickier theoretically. In the mainstream Montagovian literature on adjectives, e.g. [29, 58], the inference schema for intersective adjectives is as follows:

(40) From Adj(N)(x) *infer* N(x) $\wedge$ Adj(x).

But what does Adj(x) mean? It means that *Adj* is true of x. In the case of *black man* it means that x is a man and x is black. But what does it mean in terms inference? Well, what it means is that if we have $Adj(N_1)(x)$ and given a noun N_2, where $N_1 < N_2$, then it should follow that:

(41) From Adj(N_1)(x) *infer* N_1(x) $\wedge$ Adj(N_2)(x).

In practical terms and taking *black* as our example, this means that for every A and B where $A < B$, we have *black* $A <$ *black* B. In case no relation between A and B exists or if the subtyping relation is reversed, no inference should be possible. Indeed, given the subtyping relations and the fact that subtyping relations propagate through the constructor types, we predict the desired inferences.

Intersective adjectives can be interpreted in this respect by a simple predicate type, declaring the subtyping relations and interpreting adjectival modification as a Σ type. However, this will overgenerate for subsective adjectives. This is because in this case we have to find a way of deriving the first inference associated with intersective adjectives but not the second, thus we need to take care of the following inference only:

(42) Adj(N) $\Rightarrow$ N.

[12] This is an informal presentation here on what our correct inferences should be. In MTT semantics, given that CNs are types, this is translated to 'from $\Sigma x{:}\, [\![N]\!] \,.\, [\![Adj]\!](x)$ infer $x{:}\, [\![N]\!]$. We keep this informal notation for exemplifying the inferences we should get.

Thus, in the case of an adjective like *small* we need to predict that from *small* N, N follows but given A, where $N < A$, *small* A does not follow. To give an example, one should not be able to deduce *small animal* from *small elephant*. The solution here is to use universes in the typing for subsective adjectives, in effect having the type for subsective adjectives ranging over the universe CN. Thus, an adjective like *small* will receive the following type:

(43) $\Pi A : \text{CN}.\ (A \rightarrow Prop)$

The above idea has been proposed by [7] and it is basically an implementation of the intuition that subsective adjectives are only relevant for the particular CN they modify in each case. Thus, a small elephant is only small with respect to elephants, a skilful surgeon is only skilful for a surgeon, and so on. Using the type proposed in (43), we can have different instances of a subsective adjective, say P, depending on the choice of A, with $A : \text{CN}$. This account of intersective and non-subsective adjectives relies on the following assumptions: a) CNs are types, b) adjectives are predicates (or lower function types, see discussion in §3.2.2), c) predicates may be polymorphic. The welcome result in this approach is that inferential properties are derived via typing only and no extra axioms in the form of meaning postulates are needed.[13]

3.2 Non-subsective Adjectives

The next class of adjectives is classified into two further subclasses, that of privative and that of non-committal adjectives. Privative adjectives are those adjectives that give rise to the following inferential schema:

(44) From Adj(N) $infer$ ¬ N.

Privative adjectives include adjectives like *fake* and *former*. The privative class is potentially the most problematic class, since there is no general consensus that these adjectives do indeed give rise to such inferences. For example, for the case of *fake*, [58] has argued (convincingly in our opinion) that adjectives like *fake* are not really privative. Partee argues that privative adjectives are in fact interpreted as subsective [58, 59]. The idea is that in cases of privative modification the interpretation of the CN is coerced to include the denotations of CNs modified by privative adjectives. Thus, in the case of (45) and (46), Partee argues that the denotation of *fur* is expanded to include both *real* and *fake* furs:

(45) I don't care whether that fur is fake fur or real fur.

(46) I don't care whether that fur is fake or real.

Thus, in *fake fur*, *fur* is coerced to include fake furs as well.

A potentially similar idea was first considered, independently with [58, 59], by the second author in [40] where he proposed to use disjoint union types to represent adjectival modifications by privative adjectives, and was further studied by us in [7]. The idea is as follows: We define G_R and G_F to be the type of (real) guns and fake guns, respectively. Then,

$$G = G_R + G_F$$

represents the type of all guns. It consists of objects of the form $inl(r)$ and $inr(f)$, where $r : G_R$ and $f : G_F$. The associated injection operators $inl : G_F \rightarrow G$ are declared and $inr : G_R \rightarrow G$ as coercions:

$$G_R <_{inl} G \quad and \quad G_F <_{inr} G.$$

The following predicates can now be defined: $real\_gun$ and $fake\_gun$ of type $G \rightarrow Prop$:

$$real\_gun(inl(r)) = True \quad and \quad real\_gun(inr(f)) = False;$$

[13] One would still want to have inferences of the following form: x is a big ant ⇒ x is an ant ⇒ x is a small animal. This is correct. This information can be added as axioms assuming that CNs can be also indexed with size parameters (see the discussion on gradability). In this case, one can assume for example the following axiom: $AUNT = \Pi s, s1{:}Size.\Pi y{:}Aunt_{s1}.s > s_1.$, where s is the contextual size parameter for type $Animal$. This mean that all sizes of aunts will be less than the standard size for animals, thus they will be predicted to be small animals.

$$fake\_gun(inl(r)) = False \quad and \quad fake\_gun(inr(f)) = True.$$

It is easy to see that, for any $g : G$,

(47)$real\_gun(g)$ iff $\neg fake\_gun(g)$.

The following interpretations can be now given (both are true): for $g : G_R$:

(48)$[\![g\ is\ a\ real\ gun]\!] = real\_gun(g)$

and for $f : G_F$,

(49)$[\![f\ is\ not\ a\ real\ gun]\!] = \neg real\_gun(f)$

Given the above, it is not difficult to see that the sentences like those below can easily be interpreted as expected:

(50)Is that gun real or fake?

(51)A fake gun is not a gun.

Note that in the second example *gun* is taken to mean real gun. This needs some explanation. According to the Partee explanation, without the coercion of *gun* to include fake guns, the adjective *real* would also be redundant (since all guns would be real guns). So, in the above example we take this to mean that a fake gun is not a real gun.

Adjectives like *former* have similar problems, given that there is no consensus on the inferences they give rise to. Some people accept the judgments associated with privative adjectives while others categorize them as non-committal, i.e. as giving rise to no inference whatsoever The two options are given below:

(52)Former(N) $\Rightarrow \neg$ N.

(53)Former(N) $\Rightarrow$?.

On the assumption that *former* is a privative adjective, one can pursue an analysis similar to the one given for *fake*. However, it seems to us that in giving a correct account of *former*, one needs to take into consideration the time parameter associated with *former* and that the correct inference associated with *former* should be that from *former N* one can infer *N* in some past time. One might argue that this should also involve the inference that *former N* implies the negation of *current N* but this seems to be disputed by data like the following:

(54)The former president, which happens to be the current one as well, addressed the crowd

The relevant intuition here seems to be that *former N* implies $\neg$*current N*, i.e. that $\neg N$ holds at the current time, but only if there is a time in between the current time and the past time where *former N* was true, where $\neg N$ was the case. Thus, a former president that was re-elected after having a break from presidency can also be the current president. But, it seems that a re-elected president without any break from presidency cannot be considered a former president, but a current president. If this is true, analyses in the style of [19], i.e. a predicate modifier approach, where the noun does not hold of its argument in the evaluation time but a time preceding it overgenerate. We present a more recent version of this idea presented by [54] where worlds are substituted with Kratzerian situations [33]:

(55)$[\![former\ president]\!] = \lambda x.\lambda s.\exists s'[s' < s \wedge president(x)(s') \wedge \neg president(x)(s)]$

We want to propose a refinement of this idea in MTTs. The idea is to use CNs indexed with a time parameter. Following [8], we introduce a type $Time$ of times to deal with the parameter to time (e.g. Ranta [61] also uses a similar type to deal with tense). Over $Time$, we have a precedence relation $\leq$ as well as a specific object $[\![now]\!] : Time$, standing for 'the current time' or 'the default time'. The $\leq$ relation conforms to the usual properties, *irreflexivity*, *transitivity*, *asymmetry*, *connectedness* and *density*. $Time$ can be also specified as an inductive type, in the sense proposed in [8]. This proposal is shown below:

(56) $[\![date]\!] : DATE \rightarrow Time.$

where $DATE$ consists of the triples (y, m, d) where y ranges over integers to represent years, m over Jan to Dec to represent months, and d over the days $1, 2, ...$ to represent days.

With these assumptions in place, we can further assume that CNs are indexed with a time parameter. Thus a common noun like *mayor* is not interpreted as a plain CN but as a family of types indexed by the parameter $t{:}Time$:

(57) $mayor(t) : \textsc{cn}.$

(58) $[\![former\ mayor]\!] = \neg mayor(now) \wedge \exists t : Time.\ t < now \wedge mayor(t) \wedge \exists t_1{:}Time, t_1 \neq t \wedge t_1 \leq now \wedge \neg mayor(t_1).$

Former is defined in this respect as $[\![former]\!] : (Time \rightarrow \textsc{cn}) \rightarrow \textsc{cn}$, obtained by abstracting *mayor* in (58):

(59) $[\![former]\!](p) = \neg p(now) \wedge \exists t : Time.\ t < now \wedge p(t) \wedge \exists t_1{:}Time, t_1 \neq t \wedge t_1 \leq now \wedge \neg p(t_1).$

Similar ideas can be developed for other temporal adjectives like *past, current.*[14] Another thing we should look at is the typing. The typing associated with *former* as we have said is $[\![former]\!] : (Time \rightarrow \textsc{cn}) \rightarrow \textsc{cn}$. It involves a time parameter and also returns a CN instead of a proposition. The latter will predict that adjectives like *former* cannot be used *predicatively.*[15] However, giving this type will mean that any CN type can be combined with *former*, a fact that is not true. For example, in general non-animate CNs are not possible with *former*:

(60) # Former table/house/piano/man/human

Furthermore, cases of nouns that denote a permanent property, in effect individual level nouns, are also not good with former:

(61) # Former man/human

From these data, it seems that former combines with stage level nouns that are also subtypes of type $Human$. one might argue that the stage level restriction is in itself not strange, given that the semantics of *former* specify a change from a time where an x is true of the property and a time where x is not true any more. For CNs that denote a permanent property, e.g *man* or *human*, combination with *former* is thus incompatible because such change is inherently contradictory. It is worth noting, in support of the argument we are making here, that cases where *former* actually combines with individual level nouns, these are coerced into receiving a stage level interpretation:

(62) Her former man

In the above example *man* basically has the meaning of *boyfriend*, *husband*, in effect it is turned into a stage level noun. The way this kind of coercions precisely work is a matter well beyond the scope of this paper. However, coercive subtyping is a mechanism that has been argued to be fit for dealing with a wide range of linguistic coercions, see e.g. [41, 2]. What is relevant here, is that former needs a stage

[14] It has to be noted however that in order to do full justice to this kind of adjectives one has to engage into the issue of temporal sensitivity of nouns. In our case, this means that we have to look at the way the temporal index of CNs interacts with the rest of the sentence. Unfortunately, such engagement cannot be done in this paper for obvious reasons of space. This however as well as the more general issue of providing a solid temporal theory using MTTs is one of the things that we are currently looking at. For more information on the temporal sensitivity of CNs, the interested reader is redirected to [20, 55, 68] for thorough discussions and proposals on temporal sensitivity.

[15] Please see the remark on predicativity at the end of this section.

$$\frac{}{mayor, student... : \text{CN}_s} \qquad \frac{A{:}\text{CN}_s}{A{:}\text{CN}}$$

Fig. 3 Some introduction rules for CN_s.

level noun to combine with and in order to combine with individual level nouns, a some kind of coercion into stage level has to be performed.

In order to express this idea, we introduce the subuniverse CN_s such that $\text{CN}_s < \text{CN}$that contains the (names) of the types for stage level CNs. We follow pretty much the same idea as in the construction of the CNuniverse, namely the fact that CN_sshould be open. The most important rule is one where types in CN_swould also be in CN. In 3.2, we show two of these rules, one introducing stage level nouns into the CN_suniverse and one denoting that every name $\text{CN}_s$is also in CN.

In general, one can argue that the category of privative adjectives can be suspended. The various adjectives of this class can be in fact argued to be either subsective or non-committal rather than privative. With this remark, we can now look at the category of not committal adjectives. This category involves adjectives that do not give rise to any inference whatsoever:

(63)Adj(N) $\Rightarrow$?

Classical cases of this category include modal adjectives like *alleged* or *possible*. Let us look at the adjective *alleged*. What does *alleged N* mean? What is an *alleged murderer* for example? An alleged murderer is someone who has been alleged at least by someone that he is a murderer. The assumption we are going to make is that an alleged murderer is a murderer in someone's belief context. Then, we can formulate belief contexts in the sense of Ranta [61]: a belief context is a sequence of assumptions that an agent p has made. More precisely, the belief context of an agent p, Γ_p, is a context of the following form:

(64)$\Gamma_p = x_1 : A_1, ..., x_n : A_n$

Based on this, Ranta proposes the belief operator B_p, defined as

$$B_p A = \Pi\Gamma_p.\ A = \Pi x_1{:}A_1 ... \Pi x_n{:}A_n.\ A.$$

As a consequence, $B_p A$ is true if and only if A is true in Γ_p. Now, given the above, we can interpret *alleged* as follows for $A_N : \text{CN}$ being the interpretation of a common noun N:

(65)$[\![alleged\ N]\!] = \Sigma p{:}\,[\![human]\!]\,.\ B(p, A_N)$

where $B(p, A) = \Pi\Gamma_p.\ A$ with Γ_p being the belief context of $p : [\![human]\!]$.[16]

Again, as in the case of *former*, one should restrict the scope of *alleged*. This is because the definition in (66) will overgenerate, since it will predict combinations like *alleged chair* and *alleged democracy* to be possible. It seems that alleged, similarly to former, can only combine with stage level CNs. In this sense, we update the definition for $[\![A_N]\!] : \text{CN}_s$.

3.2.1 A note on unknown beliefs

The definition proposed for alleged involves existential quantification over an agent p: $[\![human]\!]$ and then a judgment concerning the belief context of this agent. The problem that arises here is the issue of unknown beliefs, simply put how we define belief contexts for unknown agents. There are a number of ways to deal/argue with this problem. The first would be to assume that we know beforehand the belief contexts of types $[\![human]\!]$. This would predict a kind of omniscience on our behalf, which is not very plausible. Another way at looking at the matter, would be to define *alleged* or any other word involving belief contexts and unknown agents as involving quantification not over $[\![human]\!]$ but a subclass $[\![human_B]\!]$,

[16] This is the analog of a formula that involves existential quantifications. One may turn such types into propositions by means of the following operation: for any type A, $Exists(A) = \exists x{:}A.True$. Then, with this mechanism, (66) can be represented as the proposition $\exists p{:}Human.\ Exists(B(p, A_N))$.

i.e. the collection of those humans for which their beliefs are known ($[\![human_B]\!] < [\![human]\!]$). Lastly, there is another way, which to our opinion might give a good solution to this problem but its formal details cannot be really specified here. The idea is as follows: When we talk about other people's beliefs what we are actually representing is our perspective of the other person's belief context, i.e. our representation of this context depending on the information we have. Sometimes this representation might exactly match the other person's belief context, sometimes it can be a close approximation and sometimes it might be even totally different. So when someone utters 'John believes that he opened the door', what we really get is not that the proposition $opened(the\_door)(John)$ is part of Γ_j but rather part of $\Gamma_{j(u)}$, that is John's belief context as seen by the utterer u. This is we believe an interesting idea that would be able to distinguish and predict different contexts in cases like the following:

(66) A: John believes that Mary is a left winged radical. B: No, John does not believe this.

In the above cases, we have two distinct utterers with different versions of John's belief context, $G_{j(u1)}$ and $G_{j(u2)}$. These cases would be impossible to get by assuming one G_j context. In this sense, the belief context is relativized to the utterer in each case. Thus, in cases of unknown agents, this b elief context might just be the minimal context including $B(p, A_N)$ in the case of alleged and nothing more. We take this to be a promising way to deal with these cases. However, we have to look at the formal details behind such a proposal. However, this task cannot be taken up in this paper. Similar considerations apply to the other cases where belief contexts are used in this paper.

3.2.2 A note on predicativity

It is a well-known fact that a number of adjectives cannot be used predicatively but only attributively. We have considered adjectives of this sort so far, e.g. *alleged* or *former* are of this type. An explanation for the behaviour of this class of adjectives comes from Coppock [15] who argues that adjectives that cannot be used predicatively are not semantically predicative. In standard terms, this means that the adjective is not a predicate. It has been pointed out to us that if we assume a lower predicate type for adjectives, problems arise w.r.t non-predicative adjectives. This is correct. However, we do not assume that all adjectives are predicates. We do assume that there are no higher order types associated with adjectives, but we do assume that adjectives might not be predicates. We have already exemplified the latter point for adjectives like $former$ where the typing for $former$ was $(Time \to \mathrm{CN}_s) \to \mathrm{CN}_s$. In effect, for non-predicative adjectives the idea is that these are function types that return a CN type rather than a predicate (i.e. returning an element of type $Prop$). Similar considerations apply to the adjective *alleged* and generalizing, to all non-predicative adjectives.

4 Gradable Adjectives and Multidimensional Adjectives

4.1 Gradable Adjectives

Adjectives like *small* and *large* do not only belong to the category of subsective adjectives but have a further distinguishing property, i.e. they are also gradable adjectives. In general, by gradable adjectives we mean the class of adjectives that involve some kind of grading parameter that allows them to be quantified according to it. For example, in the case of *small* and *large* the grading parameter is size. Gradable adjectives are traditionally found in comparative or superlative forms and can be further modified by grading adverbs. The usual assumption made in the literature on gradable adjectives, as already noted in the inroduction and which conforms to our intuitions about this type of adjectives, is that some kind of measurement is involved. Then, two approaches are found depending on whether this parameter is formally encoded in the definition for gradable adjectives or not. The type of a gradable adjective differs minimally from the type of a non-gradable one. Simply put, gradable adjectives have a degree argument, while non-gradable ones don't. For example, on the assumption that *small* and *large* are given lower predicate types, (i.e. $e \to t$), then the modified typing to further deal with gradability will be $d \to (e \to t)$. Example (7), showing the definition for adjective *small* under this view is repeated below:

(67) $[\![small]\!] = \lambda d{:}Degree.\lambda x{:}e.Height(x) \leq d$

Proponents of such an approach can be found in [3, 70, 26] among others. The other option for treating gradable adjectives is to assume that they involve the same typing as non-gradable ones. The difference between the two is that gradable adjectives, even though being predicates from individuals to truth values, they further involve partially ordered domains. Gradable adjectives impose a partitioning of this partially ordered domains. For objects x that fall into the upper side of the domain imposed by adjective A, $A(x)$ is true while for objects y on the lower side of the scale, $A(y)$ is false. This is the approach that Kennedy [31] calls the Vague Predicate Approach. Proponents of such an approach can be found in [35, 50, 32, 69]. The list of accounts for gradable adjectives is quite long to be fully mentioned and the interested reader is redirected to Kennedy [31] for more information on these accounts and additional references. Another nice and most recent overview of the two approaches is [34].

In what follows, we are going to propose a way of dealing with gradable adjectives in MTTs. The account is based on earlier treatments proposed by the authors as regards comparative adjectives [11] and it is in line with the approach that gradability involves an implicit grade parameter but transferred to an MTT setting. The account uses indexed types, in particular CNs that are indexed by a degree parameter. Let's consider *shorter than* as an example, taking heights to be measured by the type $[\![Height]\!]$ of numbers such as 1.70.[17] We are then led to consider the family of types $[\![human]\!] : Height \to Type$ indexed by heights: $[\![human(n)]\!]$ is the type of humans of height n. Then, *shorter than* is defined as follows:[18][19]

(68) $[\![SHORTER\_THAN]\!] : \Pi i, j : [\![Height]\!] \,.\, \Sigma p : [\![human(i)]\!] \to [\![human(j)]\!]$
$\to Prop.\forall h_1{:}\, [\![human(i)]\!]\, \forall h_2{:}\, [\![human(j)]\!] \,.\, p(h_1, h_2) \leftrightarrow i < j.$

(69) $[\![shorter\ than]\!](i, j) = \pi_1(SHORTER\_THAN(i, j)) : [\![human(i)]\!] \to [\![human(j)]\!] \to Prop$

With this account in line, one can propose that positives are just special instances of comparatives. The only difference in this case is that $<$ is between a degree provided by the $[\![human(n)]\!]$ argument and a contextually provided degree parameter. The typing in this respect is different than comparatives (missing one indexed argument), being as follows:[20]

(70) $[\![SHORT]\!] : \Pi i : [\![Height]\!] \,.\Sigma p : [\![human(i)]\!] \to Prop.\forall h_1{:}\, [\![human(i)]\!] \,.p(h_1)$
$\leftrightarrow i < n.$

(71) $[\![short]\!](i) = \pi_1(SHORT(i)) : [\![human(i)]\!] \to Prop$

A question that naturally comes to mind and has been the subject of pretty much all degree based accounts that assume a standard contextual degree parameter, is where do we get this parameter from. The idea is that the context (in the MTT sense) that the adjective is evaluated at, provides this value. More specifically, a polymorphic function $STND$ (standing for 'standard') is assumed that takes a gradable adjective as an argument and returns the contextual parameter relevant for each adjective:[21]

(72) $[\![STND]\!] = \lambda A{:}\text{CN}.\lambda i{:}D.\lambda P{:}A_i \to Prop.\exists n_1{:}Nat.n_1 = n \wedge i <> n$

[17] Here we do not spell out the type $[\![Height]\!]$. One might take $Height$ to be the type of natural numbers and use 170 to stand for 1.70, etc.

[18] The transitive properties of comparatives are not encoded in this example for reasons of simplicity. One may very well do so having as a guide the previous entry without measures.

[19] The definition involves a bi-implication, given that if the height of human x is less than the height of another human y, then it is also the case that x is shorter than y. The definition also works as an implication.

[20] Where $n{:}\, [\![Height]\!]$, the contextual degree parameter.

[21] $<>$ stands for either $<$ or $>$. For type D see the following discussion.

Now, the exact value of n (or a range of values) might be specified in the context. But sometimes it might not. For example assume that we hear out of the blue that *George is tall.* One way is to interpret this as matching the value that our knowledge context has or to assume that there is a value that we do not know yet in the context of the conversation (here we take this to be Γ):[22]

(73) $K_{sp} = [\![(STND(Height)(i)(tall)]\!] = (180 = n) \wedge i > n)$

(74) $\Gamma = [\![(STND(Height)(i)(tall)]\!] = (\exists n_1{:}Nat.n_1 = n) \wedge i > n)$

Note that the value in Γ might be elaborated via context extension. Following Ranta [61] particular, we define a mapping $f{:}\Delta \rightarrow \Gamma$, where everything that is in Γ is also in Δ plus some potentially extra information.[23] Also, this idea of using TT contexts will be natural in cases the standard value is way-off the one usually found in the default context. Consider for example the context of all statements pertaining to basketball. In there, and given the nature of the sport, the meaning of *tall* and *short* is inevitably different.[24]

Lastly, in order to deal with different types of degrees on a more general scale, we can introduce the universe of all degree types, D. All the types in this universe are totally ordered and dense, i.e. they respect the following axioms:

(75) $TR = \Pi A{:}D.\forall d, d_1, d_2{:}A.d \leq d_1 \wedge d_1 \leq d_2 \rightarrow d \leq d_2$
$ANTISYM = \Pi A{:}D.\forall d, d_1{:}A.d \leq d_1 \wedge d_1 \leq d \rightarrow d \leq= d_1$
$REFL = \Pi A{:}D.\forall d{:}A.d \leq d$
$DENS = \Pi A{:}D.\forall d, d_1{:}A.d \leq d_1 \rightarrow \exists d_2{:}A, d \leq d_2 \leq d_1$

In the universe D, one can find types $[\![Height]\!], [\![Weight]\!], [\![Width]\!]:D$ among other types.

4.2 Multidimensional Adjectives

Multidimensional adjectives are those adjectives that can be quantified across different dimensions. Such cases include adjectives like *sick*, *healthy* etc. Two different classes of multidimensional adjectives are distinguished [63], positive and negative. Basically, every positive adjective has a negative counterpart, its antonym. Thus, for the positive adjective *healthy*, we get the negative *sick.* The difference is the form of quantification over dimensions in each case. Positive adjectives involve universal quantification over dimensions, while negative adjectives existential quantification. Thus, for someone to be considered *healthy*, s/he must be healthy in all dimensions, whereas *sick*, it suffices to be sick across one dimension only. In order to see this, one can use the exception phrase *except.* This phrase is only compatible with universal quantification. As can be seen below, only *healthy* is compatible with *except*, *sick* being infelicitous:

(76) Dan is healthy except with respect to blood pressure

(77) # Dan is sick except with respect to blood pressure

This idea put forth by Sassoon can be implemented in an MTT setting using an inductive type for multiple dimensions. Let us explain. Consider an adjective like *healthy.* In order for someone to be considered healthy, one must be able to universally quantify over a number of 'health' dimensions such

22 K_{sp} stands for the speaker's knowledge context.

23 This corresponds to what Boldini [4] called logical inference between contexts. In particular $\exists n_1{:}Nat.n_1 = n$ will be reduced to $180 = n$ by the $\exists$ elimination rule.

24 Another consequence of this approach is that given the polymorphic type of the function, the n is always relativized to both A:CN and $i{:}D$. c.)Thus, for a polymorphic adjective like *small*, the contexrualized value will be relativized to the type A (e.g. $[\![human]\!], [\![animal]\!]$ etc.).

as *cholesterol, blood pressure* etc. To formalize this, we can introduce the inductive type $Health$ of type D as follows:[25]

(78)Inductive $[\![Health]\!]$: D := Heart | Blood_ pressure | Cholesterol.

We now define the adjective $Healthy$ to be of the following type, where we use $[\![human]\!]$ as a type rather than a type-valued function as used earlier:

(79) $[\![Healthy]\!] : [\![Health]\!] \to [\![human]\!] \to Prop$

We can use this as a primitive to define *healthy* and *sick* as follows:

(80) $[\![healthy]\!] = \lambda x\colon [\![human]\!] .\forall h\colon [\![Health]\!] .Healthy(h)(x)$
(81) $[\![sick]\!] = \lambda x\colon [\![human]\!] .\neg(\forall h\colon [\![Health]\!] .Healthy(h)(x))$

The idea here is to $[\![Health]\!]$ as an inductive type, in order to encode all the different dimensions we need. This is one way of dealing with multidimensional adjectives in MTTs. Of course, there are a number of issues in case one wants to further give a full theory of gradable and multidimensional adjectives. Our goal was to show an initial way of approaching these kind of adjectives in MTTs. The interested reader who wants to further investigate the issue of gradability and multidimensionality of adjectives is directed to [34] and references therein for gradability and [63] for multidimensional adjectives.

5 Adverbial Modification

The literature on adverbs in MTTs is rather poor. The only paper specifically dealing with adverbial modification is [6].[26] However, adverbs have been also treated in [11] as part of a discussion on Natural Language Inference (NLI). There, a first approach of some aspects of adverbial modification like veridicality, non-veridicality, adverbial typing and intensional adverbs among others has been attempted. Here, we extend the approach of [40, 6, 10] to further adverbial classes and deepen the analysis given there.

The first thing to discuss is the notion of typing, given that MTTs are many sorted and in general are quite different typed systems than the ones presented in different versions of simple type theory. The proposals put forth so far in the literature are based on the second author's proposal [40, 41], subsequently followed in [8, 6], according to which, VP adverbs receive a polymorphic type extending over the universe CN (82), while sentence level adverbs are just functions from propositions to propositions (83):

(82)$\Pi A : \mathrm{CN}.\ (A \to Prop) \to (A \to Prop)$
(83)$Prop \to Prop$

5.1 Veridicality

A very basic distinction in terms of the semantic properties of adverbs, in particular the inference patterns that they give rise to, concerns what has been dubbed as veridicality. Veridicality is found in both VP and sentence level adverb. Veridicality in the case of sentence adverbs means that $Adv(P)$ presupposes P whereas in the case of VP adverbs $V(P(x))$ presupposes $P(x)$).[27]

In order to take care of veridical adverbs, [11] have proposed a technique similar to the one used for comparatives. First, an auxiliary object *veridical* is defined (two definitions one for VP and one for sentence adverbs, VER_{Prop} and VER_{VP} respectively), and then veridical adverbs are defined as the first projection of this auxiliary object ($ADV_{ver-Prop}$ and ADV_{ver-VP}):

[25] The inductive type Health in CN is the finite type (also called an enumeration type), sometimes written as $\{Heart,\ Blood\_pressure,\ Cholesterol\}$.
[26] One of the reasons for this is that researchers found it difficult to give adverbial typings when CNs are interpreted as types. The first to discuss adverbial typings was Luo [40] who proposed to use the universe CN and polymorphism to solve this problem. This proposal was followed in [6] and also in this paper.
[27] With $P{:}A \to Prop$, $x{:}A$ and $A{:}\mathrm{CN}$

(84) $[\![VER_{Prop}]\!] : \Pi v : Prop.\ \Sigma p : Prop.p \supset v$

(85) $[\![VER_{VP}]\!] : \Pi A : \textsc{cn}.\Pi v : A \to Prop.\ \Sigma p : A \to Prop.\forall x{:}A.p(x) \supset v(x)$

(86) $[\![ADV_{ver-Prop}]\!] = \lambda v : Prop.\ \pi_1(VER_{Prop}(v))$

(87) $[\![ADV_{ver-VP}]\!] = \lambda A : \textsc{cn}.\lambda v : A \to Prop.\ \pi_1(VER_{VP}(A, v))$

Let us see how this approach works in more detail. We will exemplify this with the sentence adverb case (the reader can then work out the VP-adverb case). According to the above, a sentence adverb like *fortunately* will be defined as in (89), i.e. as the first projection π_1 of (84):

(88) $[\![\textsc{fortunately}]\!] : \Pi v : Prop.\ \Sigma p{:}Prop.\ p \Rightarrow v$

(89) $[\![fortunately]\!] = \lambda v : Prop.\ \pi_1(\textsc{fortunately}(v))$

The type of (89) is $Prop \to Prop$: it takes an argument $v : Prop$ and returns the first component of the pair $\textsc{fortunately}(v)$, which is also of type $Prop$. Now, let us consider the following inference:

(90)Fortunately, John went $\Longrightarrow$ John went

To see that (90) is the case, we only need to realise that the second component of $\textsc{fortunately}(v)$ is a proof of

(91)$[\![fortunately]\!](v) \Rightarrow v$

Taking v to be $[\![John\ went]\!]$, (91) is the semantic representation of (90).

Note that what we have presented here only deals with the veridical property and does not say anything further about the semantics of the adverbs in each case. In order to get into the specifics of each veridical adverb, more information will be introduced, potentially in the form of a conjunction, but this is something that we have not looked at yet.

5.2 Intensional Adverbs

Veridicality/non-veridicality, as already mentioned, is just one of the aspects associated with the meaning of adverbs. This is however not sufficient to deal with a number of other aspects of the semantics of adverbials. For example, cases of adverbs like *intentionally* or epistemic adverbs like *possibly* or *allegedly* cannot be treated in this manner. One of the reasons for this is that this type of adverbs create what we call opaque contexts. The latter type of adverbs, i.e. epistemic adverbs, creates opaque contexts for both the subject and the object, while the former, i.e *intentionally*, only for the object:

(92)Oedipus allegedly married Jocaste.

(93)Oedipus intentionally married Jocaste.

From (92), we have:

(94)Oedipus allegedly married Jocaste $\not\Rightarrow$ the son of Laius allegedly married Jocaste

(95)Oedipus allegedly married Jocaste $\not\Rightarrow$ Oedipus allegedly married his mother

On the other hand, from (93) we have:

(96)Oedipus intentionally married Jocaste $\Rightarrow$ The son of Laius intentionally married Jocaste

(97)Oedipus intentionally married Jocaste $\not\Rightarrow$ Oedipus intentionally married his mother

In order to deal with these data, the first author [6] uses the type theoretic notion of context similarly to the way used by Ranta [61] and also here in this paper.

The meaning of *intentionally* is taken to be the following: there exists an agent p, that did something x intentionally, i.e. agent p believes that he did x. This analysis faces a number of problems. It is certainly true that intentions and beliefs are related in a number of ways but it seems that approaching *intentionally* via belief contexts might lead us to problems. The most obvious problem, concerns cases like the following:

(98)I accidentally bumped into someone and I believe that I did it

One would expect this sentence not to be correct on the assumption that *accidentally* is a antonym of *intentionally*.

A better analysis of the meaning of intentionally would be something like the following: A intentionally P means that A has the intention P and furthermore fulfilled this intention, i.e. P holds. In order to formalize this, we introduce the notion of intention contexts, which represent an agent's collection of intentions.

We can represent p's intentional context as a number of judgements $x{:}A$ $(A : Prop)$ corresponding to intentions, this agent holds:[28]

(99)$\mathcal{D}_p = x_1 : A_1, ..., x_n : A_n(x_1, ..., x_{n-1})$

From this, and again following [61], we can construct a generalized intention operator by binding all the variables in $\mathcal{D}_p$

(100)$I_p A = \Pi \mathcal{D}_p.A = \Pi x_1{:}A_1 ... \Pi x_n{:}A_n(x_1, ..., x_{n-1}).A$

With these ideas in mind, one can put forward an account of *intentionally* as follows:

(101)$[\![Intentionally]\!] = \lambda x : [\![human]\!] .\lambda P : [\![human]\!] \rightarrow Prop.\ I_x(P(x)) \wedge \Gamma(P(x))$

Thus, in the case of *Oedipus intentionally married Jocaste*, we get a paraphrase that Oedipus had the intention of marrying Jocaste and he did so. In (96), we see that the x is bound. If we assume $Eq(Person, O, SoL)$ in the global context, then substituting O for x and then SoL for O, we get the following(M stands for *married* and J for *Jocaste*):

(102) $[\![Intentionally\ O\ (M(J))]\!] = I_{SoL}(M(J(SoL)) \wedge \Gamma(M(J)(SoL))$

Thus, (96) is predicted. On the other hand, in order to prove that Oedipus intentionally married his mother, we need to have $M(O, MoO)$ in the intention context of Oedipus. If we assume that the intention context of Oedipus is known and according to the standard reading of the story does not involve the aforementioned intention, then this does not follow. If we assume that Oedipus' intention context is unknown, we cannot prove it nor disprove before this information becomes available.

For cases of opaque to both the subject and the object adverbs, one needs of course a different analysis. For example, for *allegedly*, [6] proposed the following that captures the aforementioned property:

(103) $[\![allegedly]\!] = \lambda P : Prop.\ \exists p{:} [\![human]\!], B_p(P)$

One of the advantages of this type of approach is that typing between intensional and non-intensional adjectives remains the same, i.e. no indices are needed in the typing. Thus, one can get away with the unwanted consequences of including intensional typing for the adjectives that are not intensional, a well known problem for Montagovian analyses of adverbs. Another welcome result of this line of approach, is that what have been dubbed as domain adverbs, can also easily be dealt with. Let us explain. Assume a domain adjective like *botanically* or *mathematically.* Note that this is based on the assumption that domain adverbs are not veridical, i.e. that they do not presuppose the truth of the proposition they modify.[29] Thus, in a sentence like *botanically, tomato is a fruit*, it does not follow that tomato is a fruit. This basically says, that tomato is a fruit in the context of botanology. The idea is to use contexts again in order to represent the domain in each case, i.e. the context of botanology (a collection of facts pertaining to botanology), using contexts. Thus, one can use the following for the adverb *botanically* and in general for domain adverbs:

(104) $[\![botanically]\!] = \lambda P : Prop.\Gamma_B P$

[28] The same ideas discussed in the case of unknown beliefs in section 3.2.1 are also relevant for unknown intentions.

[29] We will not get into a discussion on whether this is absolutely correct and some people might have different judgments with respect to this.

5.3 Manner Adverbials

Manner adverbials are a subcategory of predicational adverbs. They are VP adverbials and constrain the way/manner the subject performs the action denoted by the VP. Classic treatments of manner adverbs usually rely on Davidsonian assumptions according to which some sort of event modification is at play. For example, an adverb like *slowly* will just say that the event under consideration is a slow event. However, this does not really capture what we want, because it is the manner of the event rather than the event itself that is slow. These considerations have led researchers to argue for the inclusion of manners in their semantic ontology as well [18, 64]. For example Schafer proposes the following for the sentence *John wrote illegibly*:

(105) $\exists e[subject(john, e) \wedge write(e) \wedge \exists m[manner(m, e) \wedge illegible(e)]]$

So how are we going to treat these type of adverbs in MTTs? The first thing to look at concerns the ontology. How are manners to be represented, if at all. The reader will remember our treatment of adjectives where the use of indexed CN types was proposed, in effect CNs indexed with degree parameters. The difference here is that modification is not about individuals but rather events. However, the idea to use indexed types seems promising in this case as well. Let us explain. First, we make the assumption that verbs subcategorize for an event argument, i.e. verbs also involve an event argument. We introduce the type $Event{:}Type$. Now assuming that a predicate will have to be of type $A \rightarrow Event \rightarrow Prop$ (with $A{:}\textsc{cn}$), a predicate modifier like e.g. a manner adverb should be of the following type:

(106) $[\![ADV_{VP}]\!] : \Pi A{:}\textsc{cn}.(A \rightarrow Event \rightarrow Prop) \rightarrow (A \rightarrow Event \rightarrow Prop)$

Of course, in order for proper event semantics to be given events need to be structured. However, this is not the scope of this paper. Here we focus on how events can be modified by manner adverbs and the idea we want to put forward is that events just like CNs can be also indexed. We can introduce a general primitive type $Manner{:}Type$. With this we can assume the family of types $Event{:}Manner \rightarrow Prop$ indexed by manners, $Event(m)$ (with $m{:}Manner$) is then the type of events of manner m. In sentences with no manner adverbs or in which there is no explicit mention on what the quality of the manner is, we can assume a contextually realized manner quality $R{:}Manner \rightarrow Prop$ which corresponds to some kind of default manner quality depending on context. In the case of manner adverbials or similar expressions, the quality of the manner is explicitly specified. Thus for *illegibly* the relevant specification would be $illegible(m)$ etc. The lexical entry for *illegibly* will be as follows:

(107) $[\![illegibly]\!] = \lambda m{:}Manner.\lambda P{:}([\![human]\!] \rightarrow Event_m \rightarrow Prop).\lambda x{:}A.$
$\lambda E{:}Event_m.P(x)(E) \wedge illeggible(m)$

Needless to say that the associated veridical inference is captured with the above entry, given that it is included as the first member of the conjunction. In effect, with this entry *he wrote* always follows from *he wrote illegibly.*

5.4 Some Notes on Other Classes of Adverbs

In this section we discuss some further issues relating adverbial modification, outlining ways in which MTT-semantics can work towards an account. A natural category of adverbs to discuss, given the context of this paper, is subject oriented adverbs, or better agent oriented adverbs given that these involve the agent and not the subject.[30] Such adverbs are traditionally looked at on a par with manner adverbs, given the existence of ambiguous readings, i.e. manner/agent-oriented, with a number of adjectives. Crucially, the accounts put forth assume a property of the subject on the basis of the truth of the proposition that the sentence without the adverb denotes. Thus, the sentence *John stupidly called Mary* means that John's

[30] Evidence from passive constructions shows that this is the case. Thus, in *the boat was sunk intentionally by the government*, the paraphrase we get is that it was intentional on behalf of the government to sink the boat, rather than it was intentional on behalf of the boat to sink itself that a subject oriented interpretation would imply. See [27, 49, 22] for more details on agent oriented adverbs.

act of calling Mary was stupid. In the literature, one finds accounts like [49] where manner adverbs are treated as arguments of the verb and agent oriented adverbs as predicate modifiers. Event based accounts treat agent oriented adverbs to involve some additional structure. For example, Rexach [62] mentions that a way to capture the difference between the two classes is to assume that, in manner adverbs, the adjective is the event only, while in the case of agent oriented adverbs, both the event and the participant are:

(108) $ADV_{manner} = \lambda P\ \lambda x\ \exists e.\ P(e,x) \wedge Adj_{ADV}(e)$
$ADV_{agent} \quad = \lambda P\ \lambda x\ \exists e.\ P(e,x) \wedge Adj_{ADV}(e,x)$

A solution in MTT-semantics, that maintains the core of the analysis of manner adverbs, will involve again indexed types. But now, instead of the type of events indexed only by manners, what we have is types of events which are indexed by humans as well as manners. Thus, $[\![human(m)]\!]$ is the type of humans with manner m.

The semantics of speaker oriented adverbs seem more difficult to grasp. Here we are going to look only at speech act adverbials like *honestly, frankly.* Such adverbs can be seen as providing commentary with respect to the utterance. In this respect, the sentence *Frankly, I do not know what to say*, roughly means *I frankly tell you that I do not know what to say.* This paraphrase dating back to [65] gives rise to a way of looking at speech act adverbs as not that different from manner adverbs. Piñon [60] provides an interesting account according to which speaker oriented adverbs make reference to individual manners of speaking. We will not go into the details of his proposal. Assuming that this be a reasonable way to look at speaker oriented adverbials, one can sketch an account in MTTs as follows.

First, the type of utterance events are indexed by utterers and manners:

$$UEvent : [\![human]\!] \to Manner \to Type,$$

that is, $UEvent(u,m)$ is the type of events with utterer u and manner m. For example, the type of *frank* can be given as follows:

$$[\![frank]\!] : \Pi u : Utterer.\Pi m : Manner.\ UEvent(u,m) \to Prop.$$

Then, the adverb *frankly* can now be given the following definition: for any $u : Utterer$, $m : Manner$ and $e : UEvent(u,m)$,

$$[\![frankly]\!](u,m,e) =_{\mathrm{df}} \lambda P : Prop.\ P \wedge frank(u,m,e)$$

and *frankly* thus defined is of the following type:

$$[\![frankly]\!] \ : \ \Pi u : Utterer\ \Pi m : Manner.\ (UEvent(u,m) \to Prop \to Prop).$$

In fact, all of such agent-oriented adverbs can be defined as above.

According to account just presented, a speech act adverb is not of type $Prop \to Prop$; instead it takes both an utterance event as well as a proposition as arguments and returns a proposition.

With this last remark, we will stop our discussion on adverbial modification, leaving a number of issues unresolved. To recap, we have shown that MTTs can provide us with a rich and expressive typed language in order to deal with a number of aspects pertaining to modification. From a theoretical point of view, we hope that we have presented arguments for using MTTs for NL semantics.

6 Modification and Inference

One attractive characteristic of MTT-semantics is that it can be seen as proof-theoretic [44]. This means that the judgments in the underlying type theories can be understood by means of their inferential roles. This latter fact constitutes MTTs a good solution w.r.t consequences that the semantics proposed in each case bring about, i.e. inference. This proof-theoretic aspect of MTTs has been the reason that these are widely implemented in computer reasoning systems, i.e. proof assistants. Proof assistant technology has gone a long way since its emergence. The proof-assistant Coq is a prime example of the advance reached in the field and a number of remarkable developments have been achieved via its use (e.g. see the proof for the four colour theorem [24]). Coq implements the Calculus of Inductive Constructions (CIC), in effect an MTT. Actually, CIC is quite close to the MTT we are using, i.e. UTT with coercive subtyping [36, 45]. The fact that Coq 'speaks' so to say an MTT, in combination with the fact that it is a powerful reasoning engine, makes Coq suitable to implement and further reason about MTT semantics. The authors have exemplified the use of Coq as a means to deal with NLI in various papers [8, 9, 13]. There, it was shown that Coq can be used as a NL reasoner and we provided an evaluation against a fragment of the FraCas test suite. Besides the various practical/computational applications that such an endeavour can lead to, there is an additional side to using Coq, which has to do with the correctness of the accounts one proposes. A correct account of a certain NL phenomenon should be able to derive all the correct consequences associated with it, while on the other hand it should not derive any unwanted consequences. This is basically to say that a correct account is judged by the number of inferences it gives rise to as well as the number of them that it does not. In our case and given the nice interaction of MTT-semantics and the associated proof-technology, we can actually check whether what the propose derives the proper inferences or not. This is what we are going to see now, by looking into the consequences that a number of our proposals made in this paper give rise to. In what follows we give a short introduction to Coq and then test the predictions of our account using MTT derivations for some cases and also presenting the relevant implementations in Coq along with explanation in the appendix.

6.1 The Coq Proof Assistant

The idea behind Coq is simple and can be roughly summarized as follows: you use Coq in order to see whether propositions based on statements previously pre-defined or user defined (definitions, parameters, variables) can be proven or not. As we have said, is a dependently typed proof-assistant implementing the calculus of Inductive Constructions (CiC, see [16]). This means that the language used for expressing these various propositions is an MTT. To give a very short example of how Coq operates let us try to prove a propositional tautology in the system, let us say the following:

(109) $((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$

Given Coq's typed nature we have to introduce the variables P, Q, R as being of type *Prop* (P, Q, R:*Prop*). Now in order to get Coq into proof mode, we have to use the command *Theorem* followed by the name we give to this theorem and then followed by the theorem we want to prove:

(110) *Theorem A*: $((P \vee Q) \wedge (P \to R) \wedge (Q \to R)) \to R$

This will put Coq into proof mode:

```
Theorem A:((P\/Q)/\(P -> R)/\(Q->R))->R.
1 subgoal
============================
(P\/Q)/\(P -> R)/\(Q -> R)->R
```

Now, we have to guide the prover to a proof using its pre-defined proof tactics (or we can define our own). For the case interested, we first introduce the antecedent as an assumption using *intro*:

```
A < intro.
1 subgoal
H :(P \/ Q)/\(P -> R)/\(Q -> R)
============================
R
```

We split the hypothesis into individual hypotheses using *destruct*

```
destruct H0.
1 subgoal
H : P \/ Q
H0 : P -> R
H1 : Q -> R
============================
R
```

Now, we can apply the elimination rule for disjunction which will basically result in two subgoals:

```
elim H.
2 subgoals
H : P \/ Q
H0 : P -> R
H1 : Q -> R
============================
P -> R
subgoal 2 is:
Q -> R
```

The two subgoals are already in the hypotheses. We can use the *assumption* tactic that matches the goal with an identical hypothesis and the proof is completed:

```
assumption. assumption.
1 subgoal
H : P \/ Q
H0 : P -> R
H1 : Q -> R
============================
Q -> R
Proof completed.
```

Now, in order to reason with NL semantics, we basically implement our theoretical work on NL semantics and then look at the consequences these have as regards inference. To give an example, consider the case of the existential quantifier *a*. Quantifiers in MTTs are given the following type, where A extends over the CN (this is reminiscent of the type used for VP adverbs):

(111) $\Pi A : \textsc{cn}.\ (A \to Prop) \to (A \to Prop)$

We provide a definition based on this type, giving rather standard semantics for the existential (in Coq notation):

```
Definition some:=fun(A:CN)(P:A->Prop)=>exists x:A,P x.
```

This says that given an A of type CN and a predicate over A, there is an $x{:}A$ such that P holds of x.

Imagine that we want to see the consequences of this definition. For example we may want to check whether *John walks* implies that *some man walks* or that *some man walks* implies that *some human walks*. We define, following our theoretical assumptions about CNs, *man* and *human* to be of type CN and declare the subtyping relation *man* $<$ *human*. This is all we need to get the above inferences. These assumptions suffice to prove these inferences in Coq.

6.2 Testing the Theory

In this section we check a number of the accounts we have provided with respect to the inferences they give or do not give rise to. At first let us see the inferences we get with respect to intersective adjectives. Traditionally, the inferences of intersective adjectives that need to be captured are the following two:[31]

(112) Adj(N)(x) $\Rightarrow$ N(x).

(113) Adj(N_1)(x) $\Rightarrow$ N_1(x) $\wedge$ Adj(N_2)(x), if $N_1 < N_2$.

The first inference is taken care of by the Σ type analysis, where the first projection is declared as a coercion. We exemplify this with the example showing *a man walks* from *a black man walks*,[32] where $[\![walk]\!] : [\![animal]\!] \rightarrow Prop$:

- $\Sigma(man, black) < [\![man]\!] < [\![animal]\!]$ (by first projection as coercion);
- therefore, $\exists x : \Sigma(man, black).\ walk(x)$ implies $\exists y : man.\ walk(y)$;
- that is, $[\![$*a black man walks*$]\!]$ implies $[\![$*a man walks*$]\!]$.

The second inference can be done in a similar way, given the rules for coercive subtyping as in [37, 45]. For example, we can infer *a black human walks* from *a black man walks*.

However, encoding this in Coq is trickier. This has to do with a defect in Coq's coercive subtyping mechanism. In a nutshell, Coq does support subtyping, but however does not allow subtyping to propagate through the constructors, i.e. in our case it does not allow the inference $\Sigma([\![man]\!], [\![black]\!]) < \Sigma([\![human]\!], [\![black]\!])$ given $[\![man]\!] < [\![human]\!]$.[33] Thus, one has to declare individual coercions between the Sigma types, e.g. $\Sigma([\![man]\!], [\![black]\!]) < \Sigma([\![human]\!], [\![black]\!])$. However, once we do that an additional problem arises: that of ambiguous paths. Given that $\Sigma([\![man]\!], [\![black]\!]) < [\![man]\!]$, $\Sigma([\![human]\!], [\![black]\!]) < [\![human]\!]$ there are two paths from $\Sigma([\![man]\!], [\![black]\!])$ to $[\![human]\!]$, one via $\Sigma([\![man]\!], [\![black]\!]) < [\![man]\!] < [\![human]\!]$, and one via $\Sigma([\![man]\!], [\![black]\!]) < \Sigma([\![human]\!], [\![black]\!]) < [\![human]\!]$ and again this is not allowed. This is a rather serious flaw in the subtyping mechanism that does not allow to evaluate our account at this point. Note that the proof-assistant Plastic that implements UTT does not give rise to this problem. In order to sidestep this, we basically introduce a hack, according to which *black man* has two senses: on one sense, it is associated with a coercion to *man*, while on the other, with a coercion to *black human*. If we do this, the inference we want goes through and we can prove things like the following:[34]

(114) A black man walks $\Rightarrow$ A black human walks

Note that this problem of inference that had to be sidestepped in this way is only present in the attributive use of the adjective, i.e. when a Σ type is involved. In predicational uses, this problem in inference does not arise and examples like the following are predicted without any hacks in the Coq code:[35]

(115) Some man is black $\Rightarrow$ Some human is black

Turning to subsective adjectives, we need to capture the fact that:

(116) from Adj(N) $infer$ N.

The polymorphic type we proposed for adjectives really takes care of this, since it will assume that the inference is valid for the specific class in each case. The inference from Adj(N) is captured via the first projection coercion, no surprises here. Now, one will want to see that an inference of the following short is not valid for subsective adjectives but it is true for intersective adjectives:

(117) some x Adj $\Rightarrow$ some y Adj (where $x < y$)

[31] Again, note that in MTTs, $N(x)$ will be the type judgment, $x{:}N$.

[32] This is easily proven in Coq (we use dependent record types for encoding the Σ type approach). See the appendix for the actual code and example.

[33] Such propagations of suutyping relations through type constructors is desirable, but it is not implemented in Coq.

[34] See appendix for the actual code.

[35] See appendix for the code and example

Indeed, for intersective adjectives, this is the case. For example assuming *black* is of type $object \rightarrow Prop$, the above can be proven. However, given the typing we have proposed for subsective adjectives, no proof is found. For if we try to prove *George is a small animal* from *George is a small man* we are stuck since we are basically trying to prove $small(animal)(George)$ from $small(man)(George)$. Given that *small* is relativized to different domains in each case, it seems that no proof can be found.

With privative adjectives like *fake* and assuming an analysis as this was sketched here where these involve coercion of the CN to include fake CN denotations, we proposed a disjoint union type. Using this type we can predict that a fake gun is a gun but it is not a real gun. For example one can prove that all real guns are not fake guns and vice versa in Coq

The account put forth for degree adjectives has already been checked using Coq in [8] for the comparative cases. It is rather straightforward to extend it to the positive case. We will not do it here for reasons of space. We will rather look at the case of multidimensional adjectives. The account proposed here can be straightforward encoded in Coq by defining an inductive type $Health$ in the same sense we did in 78:[36]

```
Inductive Health:CN:=Heart|Blood_pressure|Cholesterol|...
```

Then following the ideas sketched in this paper for multidimensional adjectives one can derive the following in Coq (see the appendix for the code):

(118) John is sick $\Rightarrow$ John is not healthy

(119) John is not healthy with respect to one health dimension $\Rightarrow$ John is sick

(120) John is healthy with respect to one health dimension $\not\Rightarrow$ John is healthy

Similarly, for adverbs we can also check the accounts proposed. We show this by giving the example of veridical VP adverbs. For example, if we define *slowly* as in (87), the inferences like the one shown below follow:

(121) John walks slowly $\Rightarrow$ John walks

since, by (87), one obtains from the second projection that *walk slowly* implies *walk*.

As an additional example, consider the analysis we have provided for manner adverbs in this paper. We can implement this in Coq. By doing so, we can get inferences like:

(122) John writes illegibly $\Rightarrow$ John wrote in an illegible manner.

We have not yet tried the intensional cases of adverbs. We leave this as future work, even though we believe that this will not be difficult to do. In particular, Coq's Local section mechanism (in effect local contexts) would be useful for implementing the account of domain adverbials. But as we have said, we end the discussion here, leaving these issues for future research.

References

1. N. Asher. *Lexical Meaning in Context: a Web of Words.* Cambridge University Press, 2012.
2. N. Asher and Z. Luo. Formalisation of coercions in lexical semantics. *Sinn und Bedeutung 17, Paris*, 223, 2012.
3. R. Bartsch and T. Vennermann. *Semantic Structures.* Athenaum, 1973.
4. P. Boldini. Formalizing contexts in intuitionistic type theory. *Fundamenta Informaticae*, 4(2), 2000.
5. P. Boldini. The reference of mass terms from a type-theoretical point of view. Paper from the 4th International Workshop on Computational Semantics, 2001.
6. S. Chatzikyriakidis. Adverbs in a modern type theory. In N. Asher and S. Soloviev, editors, *Proceedings of LACL2014. LNCS 8535*, pages 44–56, 2014.
7. 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.

[36] The '...' is not part of the actual code. It just says that more health dimensions can be added depending on the fine-grainedness we want to achieve.

8. S. Chatzikyriakidis and Z. Luo. Natural language inference in Coq. *Journal of Logic, Language and Information*, 23(4):441–480, 2014.
9. S. Chatzikyriakidis and Z. Luo. Natural language reasoning using proof-assistant technology: Rich typing and beyond. In *Proceedings of EACL2014*, 2014.
10. S. Chatzikyriakidis and Z. Luo. Natural language inference in Coq. *J. of Logic, Language and Information*, 23(4), 2014.
11. S. Chatzikyriakidis and Z. Luo. Natural language inference in Coq. *J. of Logic, Language and Information.*, 23(4):441–480, 2014.
12. S. Chatzikyriakidis and Z. Luo. Using signatures in type theory to represent situations. *Logic and Engineering of Natural Language Semantics 11. Tokyo*, 2014.
13. S. Chatzikyriakidis and Z. Luo. Logical reasoning based on formal semantics in modern type theories. Manuscript submitted to ACL2015, 2015.
14. A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5(1), 1940.
15. Elizabeth Coppock. *The logical and empirical foundations of Baker's Paradox.* ProQuest, 2008.
16. Coq 2007. *The Coq Proof Assistant Reference Manual (Version 8.1), INRIA.* The Coq Development Team, 2007.
17. D. Davidson. Compositionality and coercion in semantics: The semantics of adjective meaning. In N. Rescher, editor, *The Logical Form of Action Sentences*, pages 81–95. University of Pittsburgh Press, 1967.
18. S. Dik. The semantic representation of manner adverbials. In A. Kraak, editor, *Linguistics in the Netherlands 19721973.* Assen: Van Gorcum, 1975.
19. D.R Dowty. *Introduction to Montague Semantics.* Springer, 1981.
20. M. Enc. *Tense without scope: an analysis of nouns as indexicals.* PhD thesis, University of Wisconsin-Madison, 1981.
21. Thomas Boyden Ernst. *The syntax of adjuncts*, volume 96. Cambridge University Press, 2002.
22. Wilhelm Geuder. Oriented adverbs: Issues in the lexical semantics of event adverbs. 2000.
23. J.-Y. Girard. Une extension de l'interpretation fonctionelle de gödel à l'analyse et son application à l'élimination des coupures dans et la thèorie des types'. *Proc. 2nd Scandinavian Logic Symposium.* North-Holland, 1971.
24. G. Gonthier. A computer-checked proof of the four-colour theorem. http://research.microsoft.com/ gonthier/4colproof.pdf, 2005.
25. I. Heim. Degree operators and scope. In *Proceedings of SALT*, volume 10, pages 40–64, 2000.
26. I. Heim. Degree operators and scope. In *Proceedings of SALT*, volume 10, pages 40–64, 2000.
27. R. Jackendoff. *Semantic Interpretation in Generative Grammar.* MIT press, 1972.
28. R. Kahle and P. Schroeder-Heister, editors. *Proof-Theoretic Semantics.* Special Issue of Synthese, 148(3), 2006.
29. H. Kamp. Formal semantics of natural language. In E. Keenan, editor, *Two Theories about Adjectives*, pages 123–155. Cambridge University Press, 1975.
30. Hans Kamp and Barbara Partee. Prototype theory and compositionality. *Cognition*, 57(2):129–191, 1995.
31. Christopher Kennedy. *Projecting the adjective: The syntax and semantics of gradability and comparison.* Routledge, 1999.
32. Ewan Klein. A semantics for positive and comparative adjectives. *Linguistics and philosophy*, 4(1): 1–45, 1980.
33. A. Kratzer. An investigation of the lumps of thought. *Linguistics and philosophy*, 12(5):607–653, 1989.
34. D. Lassiter. Adjectival modification and gradation. In S. Lappin and C. Fox, editors, *Wiley-Blackwell Handbook of Contemporary Semantics.* Blackwell, 2014.
35. D. Lewis. General semantics. *synthese*, 22:18–67, 1970.
36. Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science.* Oxford University Press, 1994.
37. Z. Luo. Coercive subtyping in type theory. *CSL'96, LNCS'1258*, 1997.
38. Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1):105–130, 1999.

39. Z. Luo. Type-theoretical semantics with coercive subtyping. *Semantics and Linguistic Theory 20 (SALT20), Vancouver*, 2010.
40. Z. Luo. Adjectives and adverbs in type-theoretical semantics. Notes, 2011.
41. Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. *Logical Aspects of Computational Linguistics (LACL'2011). LNAI 6736*, 2011.
42. Z. Luo. Common nouns as types. In D. Bechet and A. Dikovsky, editors, *Logical Aspects of Computational Linguistics (LACL'2012). LNCS 7351*, 2012.
43. Z. Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513, 2012.
44. 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.
45. Z. Luo, S. Soloviev, and T. Xue. Coercive subtyping: theory and implementation. *Information and Computation*, 223:18–42, 2012.
46. C. Maienborn and M. Schafer. Adverbs and adverbials. In C. Maienborn, K. von Heusinger, and P. Portner, editors, *Semantics. An international handbook of natural language meaning*, pages 1390–1420. Mouton: De Gruyter, 2011.
47. P. Martin-Löf. An intuitionistic theory of types: predicative part. In H.Rose and J.C.Shepherdson, editors, *Logic Colloquium'73*, 1975.
48. P. Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, 1984.
49. Sally McConnell-Ginet. Adverbs and logical form: a linguistically realistic theory. *Language*, pages 144–184, 1982.
50. Sally McConnell McConnell-Ginet. *Comparative constructions in English: A syntactic and semantic analysis.* 1973.
51. R. Montague. English as a formal language. In B. Visentini, editor, *Linguaggi nella Societ e nella.* Milan: Edizioni di Comunit,l, 1970.
52. R. Montague. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Languages*, 1973.
53. R. Montague. *Formal Philosophy.* Yale University Press, 1974. Collected papers edited by R. Thomason.
54. M. Morzyci. Modificationi. to appea rin the series series Key Topics in Semantics and Pragmatics, Cambridge University Press, 2014.
55. Renate Irene Musan. *On the temporal interpretation of noun phrases.* PhD thesis, Massachusetts Institute of Technology, 1995.
56. B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction.* Oxford University Press, 1990.
57. Geoffrey Nunberg. Transfers of meaning. *Journal of Semantics*, 12(2):109–132, 1995.
58. B. Partee. Compositionality and coercion in semantics: The semantics of adjective meaning. In G. Bouma, I.M Krämer, and J. Zwarts, editors, *Cognitive foundations of interpretation*, pages 145–161. Royal Netherlands Academy of Arts and Sciences,, 2007.
59. B. Partee. Privative adjectives: Subsective plus coercion. In R. Bauerle and U. Reyle, editors, *Presuppositions and Discourse: Essays Offered to Hans Kamp*, pages 123–155. Emerald group publishing, 2010.
60. C. Piñón. Speech-act adverbs as manner adverbs. Ms. Univesity of Lille. 2013.
61. A. Ranta. *Type-Theoretical Grammar.* Oxford University Press, 1994.
62. J.G Rexach. *Questions and generalized quantifiers.* Springer, 1997.
63. G.W. Sassoon. A typology of multidimensional adjectives. *Journal of semantics*, 2012.
64. M. Schäfer. Resolving scope in manner modification. In *Empirical issues in syntax and semantics*, volume 7, pages 351–372. 2008.
65. P.A. Schreiber. Style disjuncts and the performative analysis. *Linguistic Inquiry*, pages 321–347, 1972.
66. G. Sundholm. Constructive generalized quantifiers. *Synthese*, 79(1):1–12, 1989.

67. R.H. Thomason and Stalkaner R.C. A semantic theory of adverbs. *Linguistic Inquiry*, 4(2):195–220, 1973.
68. J. Tonhauser. A dynamic semantic account of the temporal interpretation of noun phrases. In *Semantics and Linguistic Theory*, pages 286–305, 2002.
69. Johan Van Benthem. *The logic of time: a model-theoretic investigation into the varieties of temporal ontology and temporal discourse*, volume 156. Springer Science & Business Media, 2012.
70. Arnim Von Stechow. Comparing semantic theories of comparison. *Journal of semantics*, 3(1):1–77.

A The examples in Coq

The first case involves the inference a black man ⇒ a black human.

The following are needed:

```
Definition CN:=Set.
Parameter Man Human Object: CN *declaring the base types*
Axiom mh:Man->Human. Coercion mh:Man >-> Human. *subtyping*
Axiom ho:Human->Object. Coercion ho:Human>-> Object. *subtyping*
Definition some:= fun A:CN, fun P:A->Prop=> exists x:A, P(x). *quantifier*
Parameter black:object->Prop.
```

What we want to prove is:

```
1 subgoal
 ============================
some Man (fun x : Man => black x) -> some Human (fun x : Human => black x)
```

Unfolding the definition for the quantifier and using *intro*, we move the antecedent as a premise:

```
H: exists x : Man, black x
============================
exists x : Human, black x
```

Then, the *destruct* tactic first performs *elim*, it finds the appropriate destructor, applies it and then uses *intro*. The result is:

```
x : Man
H : black x
============================
exists x0 : Human, black x0
```

What we need to do is substitute x for $x0$. Applying *assumption* which matches a goal with a premise finishes the proof. However, as we have said for the attributive case, we need a hack, since in Coq, subtyping does not propagate through the constructors. So in order to prove *a black human walks* from *a black man walks* some additional coding is needed as explained in 6.2:

```
Record Blackman:CN:=mkBlackman{l4:> Man; _ :Black l4 }.
Record Blackhuman:CN:=mkBlackhuman{l5:>Human; _ :Black l5 }.
Record Blackman1:CN:=mkBlackman1{c4:>Blackhuman; _ :Black c4 }.
Inductive OneBlackman:Set:=BlackMan.
Definition Br1 (r:OneBlackman):CN:=Blackman. Coercion Br1:OneBlackman>->CN.
Definition Br2 (r:OneBlackman):Set:=Blackman1. Coercion Br2:OneBlackman>->Sortclass.
```

Moving on to subsective adjectives, we want to check whether we can infer *George is a large animal* from *George is a large man*. First the code needed:

```
Parameter Animal:CN *declaring the base types*
Axiom ha:Human->Animal. Coercion ha:Human >-> Animal. *subtyping*
Axiom ao:Animal->Object. Coercion ao:Animal >->Object. *subtyping*
Parameter George:Man.
Parameter large:forall A:CN, A -> Prop. *polymorphic type*
```

We formulate the example:

```
1 subgoal
============================
Large Man George -> Large Animal George
```

We introduce the antecedent as a premise. But it seems that there is nothing we can do besides that:

```
1 subgoal
H : Large Man George
============================
Large Animal George
```

The polymorphic nature of *Large* does not seem to allow the inference to go through.

As regards inference with *fake*, the following are needed:

```
Parameter Gun_r:CN.
Parameter Gun_f:CN.
Definition GUN:= (sum Gun_r Gun_f).
Axiom gun1:Gun_r-> GUN. Coercion gun1: Gun_r>-> GUN.
Axiom gun2:Gun_f-> GUN. Coercion gun2: Gun_f>->GUN.
Fixpoint fake (A : CN) (x : GUN):= match x with| inl _ => False | inr _ => True end.
Fixpoint real (A : CN) (x : GUN):= match x with| inl _ => True | inr _ => False end.
```

With these, we can prove that all guns, if they are real they are not fake and vice versa. We first use cbv which applies all reduction rules possible and then *intuition*, which finds all first order intuitionistic logic tautologies:

```
Coq < Theorem GUN1: forall fr: GUN, real GUN fr<-> not(fake GUN fr).
GUN1 < cbv. intuition.
1 subgoal
Proof completed.
```

For inference with veridical adverbs like *slowly*, we need the following:

```
Parameter VER:forall(A:CN)(v:A-> Prop),sigT  (fun p : A -> Prop
=>forall x:A,p x->v x).
Definition slowly:= fun A:CN, fun v:A >Prop=> projT1 (VER(A,v)).
```

Let us say we want to prove that John walks from John walks slowly:

```
1 subgoal
============================
slowly walk John -> walk John
```

We unfold the definitions and use *destruct* for the auxiliary object:

```
1 subgoal
x : Animal -> Prop
w : forall x0 : Animal, x x0 -> walk x0
============================
x John -> walk John
```

We use *apply w* and the proof is completed.

Lastly, for multidimensional adjectives, we provide the code only. The reader is encouraged to try the examples.

```
Inductive Health: CN:= Heart|Blood|Cholesterol..
Definition Sick:=fun y : Human => ~ (forall x : Health, healthy x y).
Definition Healthy:= fun y : Human => forall x : Health, healthy x y.
```

The same goes for the inferences regarding manner adverbs:

```
Parameter Manner: Set.
Parameter illegible : Manner -> Prop.
Inductive EVENT : Manner -> Type :=  EVENT1 : forall m : Manner, EVENT m.
Parameter m:Manner.
Parameter E:Event m.
ParameterWALK : Human -> EVENT m -> Prop.
Parameter M : EVENT m.
Definition illegibly:=fun (A:CN)(P:A->EVENT m->Prop)(x:A)(E:EVENT m)
=>P x E/\illegible m.
```