Language and its commonsense: where logical semantics went wrong, and where it can (and should) go

WALID S. SABA

Astound
111 Independence Drive, Menlo Park, CA 94025

The purpose of this paper is twofold: (*i*) we will argue that logical semantics might have faltered due to its failure in distinguishing between two fundamentally very different types of concepts, namely *ontological concepts*, that should be types in a strongly-typed ontology, and *logical concepts*, that are predicates corresponding to properties of and relations between objects of various ontological types; and (*ii*) we will show that accounting for these differences amounts to the integration of lexical and compositional semantics in one coherent framework, and to an embedding in our logical semantics of a strongly-typed ontology, an ontology that reflects our commonsense view of the world and the way we talk about it in ordinary language. In such a framework, as will be demonstrated, a number of challenges in the semantics of natural language can be adequately and systematically treated.

1. Introduction

In the concluding remarks of *Ontological Promiscuity* Hobbs (1985) made what we believe to be a very insightful observation: given that semantics is an attempt at specifying the relation between language and the world, if "one can assume a theory of the world that is isomorphic to the way we talk about it … then semantics becomes nearly trivial". But how exactly can we rectify our logical formalisms so that semantics, an endeavor that has occupied the most penetrating minds for over two centuries, can become (nearly) trivial, and what exactly does it mean 'to assume a theory of the world' in our semantics?

In this paper we hope to provide answers for both questions. First, we believe that a commonsense theory of the world can (and should) be embedded in our semantic formalisms resulting in a logical semantics grounded in commonsense metaphysics. Moreover, we believe that the first step to accomplishing this vision starts by rectifying what we think was a crucial oversight in logical semantics, namely the failure to distinguish between two fundamentally different types of concepts: (*i*) *ontological* concepts, that are **types** in a strongly-typed ontology; and (*ii*) *logical* concepts, that are **predicates** corresponding to properties of and relations between objects of various ontological types. By embedding ontological types in our predicates type unification and other type operations can then be used to 'uncover' missing information – information that is never explicitly stated in everyday discourse, but is often implicitly assumed as *shared* background knowledge.

In the next section we briefly discuss the phenomenon of the 'missing text', which we argue is behind most challenges in the semantics of natural language. We will then discuss the difference between ontological and logical concepts, and how acknowledging this difference effectively translates into embedding a commonsense theory of the world into our logical formalism[1]. In doing this we will review other attempts in the literature to introduce typing into logical semantics and briefly discuss how our proposal is paradigmatically different from all others. Subsequently, it will be demonstrated how differentiating between logical and ontological concepts can indeed make semantics become 'nearly' trivial: (*i*) by first suggesting how type unification over predicates embedded with various ontological types can help us 'uncover' missing text that is implicitly assumed as shared background knowledge; (*ii*) by suggesting a simple solution to two longstanding semantic puzzles; and (*iii*) by showing how a number of challenges in the semantics of natural language (e.g., lexical disambiguation, metonymy, co-predication, etc.) can be adequately and *uniformly* treated; finally, and before some concluding remarks, we will discuss in some detail the nature of the strongly-typed ontological structure that will be assumed throughout the paper. In particular, we will argue that this ontological structure is *shared*; and thus it cannot be 'invented' but should be 'discovered'. We will then make some suggestions on how this might be accomplished.

2. The Phenomenon of the Missing Text

Perhaps for computational effectiveness, as Givon (1984) once suggested, in using ordinary spoken language to express our thoughts we tend to do so by using the least possible effort; by, for one thing, uttering the least number of words that are needed to convey a particular thought. Thus, for example, we make statements such as:

(1) a. *Simon is a rock.*
b. *The corner table wants another beer.*
c. *Sheba is articulate.*
d. *Jon works in a car factory.*
e. *Carlos likes to play bridge.*
f. *BBC has a reporter in every country.*
g. *Kim asked Mary if she enjoyed the movie.*
h. *Barcelona was calm as it voted for independence.*

In our opinion, speakers (hearers) of ordinary language utter (understand) these sentences to convey (mean) the following, respectively:

(2) a. *Simon is* [**as solid as**] *a rock.*
b. *The* [**person at the**] *corner table wants another beer.*
c. *Sheba is* [**an**] *articulate* [**person**].
d. *Jon works in a car* [**−producing**] *factory.*
e. *Carlos likes to play* [**the card game**] *bridge.*
f. *BBC has a* [**different**] *reporter in every country.*
g. *Kim asked Mary if she* [**Mary**] *enjoyed* [**watching**] *the movie.*
h. [**The city of**]*Barcelona was calm as it* ['**s residents**] *voted for independence.*

[1] It should be noted that throughout this paper reference to a 'commonsense theory of the world' is a reference to the commonsense knowledge required in the process of language understanding (as performed by a 5-year old, say), and *not* to specialized domain knowledge, that one would need in general problem solving.

Clearly, any viable semantic formalism must somehow account for (i.e., somehow 'discover') this [**missing text**], as such sentences are quite common and are not at all exotic, or farfetched. In this regard, we are in total agreement with Levesque (2011), who states that in order to comprehend such sentences, "you need to have background knowledge that is not expressed in the words of the sentence to be able to sort out what is going on … And it is precisely bringing this background knowledge to bear that we informally call *thinking*" (emphasis in original)[2].

Traditionally, linguists and semanticists have dealt with such sentences by investigating various phenomena such as metaphor (2a); metonymy (2b); textual entailment (2c); nominal compounds (2d); lexical ambiguity (2e), quantifier scope ambiguity (2f); reference resolution and salient meanings (2g); and copredication (2h), to name a few. However, and although they seem to have a common denominator, it is somewhat surprising that in looking at the literature one finds that these phenomena have been studied quite independently and in many cases with incompatible proposals that are individually tailored to a specific phenomenon. In logical semantics, however, there has been quite a bit of work in recent years to develop type-theoretic semantics to deal with some instances of this phenomenon. For example, in a series of papers (see e.g. Lou, 2011; Lou, 2011; Lou 2012;) Lou introduces a type system based on Martin-Lof's type theory (Martin-Lof, 1984) where common nouns are considered to be types, and where it is shown how the machinery of *type coercion* can in such a system handle lexical disambiguation and accommodate the selection of the right sense of a structured object (see Pustojosky, 1995), that can be predicated in different ways in the same context (*copredication*). Structured objects (or a dot-type), purportedly occurs when various aspects of an object are said to be referenced in the same context, as for example in (3), where one aspect of 'book' is the PHYSICAL OBJECT sense (when being bought) while another sense is the INFORMATIONAL CONTENT aspect of the book (when being read):

(3) *John bought and read the latest book on deep learning*

While we are sympathetic to the general approach of Lou, we believe that copredication and lexical ambiguity are in fact part of the same and much simpler phenomenon (which we will shortly get into), and thus we believe that copredication and type coercion introduce complex machinery unnecessarily, not to mention that type shifting/coercing will not always produce the desired results. The same observation can be made about the work of Asher and Pustejovsky (2012) where complex machinery that permits type shifting is used to access different aspects (senses) of a structured object (a dot-type) using lexical constraints available in the context. The problem we have with this approach is that the notion of the dot-type does not seem to be cognitively plausible, not to mention that, in theory, language allows us to pick many aspects of a given object that cannot *a priori* be defined as part of the lexical semantics, but must be dynamically (compositionally) figured out (we will see this in some detail below). Moreover, and more specific to the work of Lou, we argue that there is in fact a technical problem in assuming that the entire class of common nouns should constitute the types in the system. For example, while nouns such as 'man' and 'bank' can reasonably be treated as types, this will not do in situations where the common noun is a role noun (e.g., teacher), as illustrated in (3).

---

[2] The only plausible explanation to fact that we all tend to leave out the same information is that the background knowledge needed to comprehend ordinary spoken language must be *shared*.

(3) *John is an excellent teacher*

As is obvious in (3), there is clearly a problem with taking common nouns as the types in the system since ‘excellent’ in (3) does not, on commonsense grounds, describe *John*, but his teaching (**activity**) – formally, *John is excellent* does not follow from *John is an excellent teacher*, and thus it is the hidden teaching activity that is the ontological type here, and not the surface structure common noun, ‘teacher’[3].

Starting with (Asher, 2008, Asher, 2011) and more recently in (Asher, 2015), Asher has also developed over a few years a type system that is aimed at tackling type coercion (as a model for selectional restrictions, in, for example, lexical disambiguation), as well as to handle situations involving copredication. While the same reservations we have regarding the approach taken by Lou (2012) more or less apply to the earlier work of Asher, the more recent Asher (2015) however correctly highlights the technical problems in simply performing type shifts (or type coercion), and in particular in examples such as those in (4) (from Asher, 2015).

(4) *Julie enjoyed a book. It was a mystery*

If a straightforward type shifting is performed on the first sentence, so that the type constraints imposed by ‘enjoy’ (which expects an eventuality) are satisfied, then the subsequent sentence cannot be correctly interpreted as we would have *lost*, so to speak, the physical object sense of book that is the obvious referent of ‘it’. Asher concludes, and correctly so, in our opinion, that it is not type shifting of book that must occur, but that some process in predicate composition must occur. That process, which Asher (2015) calls ‘transformation’, is essentially a functor (a function object, something similar to a lambda expression, in computing lingo), that ‘picks-up’ the desired object that can semantically link to the verb’s argument. We are in general agreement with the spirit of this approach (as nothing else will work, for one thing), but we have two reservations. For one thing, this ‘transformation’ operation is not very clear, especially in how it pick-ups *different kinds of objects*, for example, an eventuality (that is a ‘reading’) in (5) and ‘informational content’ in (6):

(5) *Julie enjoyed the book.*
$\Rightarrow$ *Julie enjoyed* **reading** *the book.*
(6) *Julie criticized the book.*
$\Rightarrow$ *Julie criticized* **the content of** *the book.*

Moreover, and if types are meant to be a set of general categories (such as PHYSICAL-OBJECT, INFORMATIONAL-CONTENT, ANIMATE, etc.), it is not clear how it can be determined in (7) that it was Barcelona’s **residents** who voted for independence (7a), that it was Barcelona’s **team** that lost to Real Madrid (7b), and that it was Barcelona’s **governing body** that announced a curfew (7c).

(7) *Barcelona was calm after it* { a. voted for independence; b. lost to Real Madrid; c. announced a curfew }

[3] The same argument applies to situations where the common noun corresponds to events, states, properties, etc. (more on this below)

The reason the highly general (and somewhat ad-hoc) type system is potentially problematic is that all the [**missing terms**] in the examples in (7) are, at a high level, of a similar type, namely some 'group of people' and thus for our pragmatics to work it must operate at a much more granular level, as we will see below.

To summarize, we want to point out that the recent efforts to incorporate type-theory in compositional semantics, thus integrating lexical semantics (and consequently pragmatics and commonsense metaphysics) with compositional semantics is a welcoming trend, and the pioneering work of Pustejovsky (2012), Lou (2011) and Asher (2015) are efforts in the right direction. In this paper we will however offer an alternative approach that we believe will solve some of the remaining challenges, as well as suggest a simple framework that uniformly and systematically handles lexical disambiguation, copredication, metonymy, etc. Here's a very brief summary of the difference between the cited proposals and ours: what is lacking, in our opinion, is not a typing scheme that allows us to combine lexical and compositional semantics (thus integrating pragmatics and commonsense metaphysics in our logical semantics). In our view, lexical and compositional semantics are already integrated, and our language already uses a typed ontological structure. The problem, however, is that we have been using predication in our logical formalisms to represent both, genuine predicates, as well as types, and it is precisely this subtle oversight that is at the source of many 'apparent' challenges in logical semantics. As this is the central point of our proposal, we shall spend a good part in the remaining of this paper to support this claim.

3. Ontological vs. Logical Concepts

We begin by making the case for a very crucial aspect to our proposal, namely distinguishing between *ontological concepts* (that are **types** in a strongly-typed ontology) and *logical concepts* (that are **predicates** that represent the properties of and the relations that hold between objects of various ontological types).

3.1 Types vs. Predicates

In *Types and Ontology* Fred Sommers (1963) suggested that there is a strongly typed ontology (that he termed 'the language tree') that seems to be implicit in all that we say in ordinary spoken language, where two objects x and y are considered to be of the same type *iff* the set of monadic predicates that are significantly (that is, truly or falsely but not absurdly) predicable of x is equivalent to the set of predicates that are significantly predicable of y. Thus, for example, while the noun phrases in (6) make reference to four distinct sets of objects, for an ontologist interested in the relationship between ontology and language, the noun phrases in (6) are ultimately referring to two types only, namely Cat and Number:

(6) a. *an old cat*
b. *a black cat*
c. *an even number*
d. *a prime number*

In other words, whether we make a reference to an *old cat* or to a *black cat*, in both instances we are ultimately speaking of objects that are of the same type; and this, according to Sommers, is a reflection of the fact that the set of monadic predicates in our natural language that are significantly predicable of old cats is *exactly* the same set that is significantly predicable of black cats (or, whatever can sensibly be said of *black cats* can also be sensibly said of *old cats*, and vice versa). In this sense, a concept such as *old* is a **predicate** that happens to be predicable of a concept such as Cat, which corresponds to a **type** in a strongly-typed ontology[4]. As such, we take the proper logical representation for the noun phrase in (7) to be that in (7b), and not the one in (7a).

(7) $[\![\textit{an adorable cat}]\!]$
a. $\Rightarrow \lambda P[(\exists x)(\text{CAT}(x) \wedge \text{ADORABLE}(x) \wedge P(x))]$
b. $\Rightarrow \lambda P[(\exists x :: \mathsf{Cat})(\text{ADORABLE}\ (x) \wedge P(x))]$

That is, 'an adorable cat' refers to some object of type Cat, a cat that, presumably, is ADORABLE. Note also that abstract objects, such as events, states, properties, etc. are also types in the ontology and can also be predicated, as shown in (8).

(8) $[\![\textit{an imminent event}]\!] \Rightarrow \lambda P[(\exists x :: \mathsf{Event})(\text{IMMINENT}(x) \wedge P(x))]$
$[\![\textit{an idle state}]\!] \Rightarrow \lambda P[(\exists x :: \mathsf{State})(\text{IDLE}(x) \wedge P(x))]$
$[\![\textit{a desirable property}]\!] \Rightarrow \lambda P[(\exists x :: \mathsf{Property})(\text{DESIRABLE}(x) \wedge P(x))]$

In our representation, therefore, we assume a Platonic universe that includes everything we talk about in our language, and where concepts belong to two quite distinct categories: (*i*) ontological concepts, such as Animal, Substance, Entity, Artifact, Book, Event, State, etc., which are types in a subsumption hierarchy, and where the fact that an object of type Human is (ultimately) an object of type Entity is expressed as $\mathsf{Human} \sqsubseteq \mathsf{Entity}$; and (*ii*) logical concepts, such as FORMER, OLD, IMMINENT, BEAUTIFUL, etc., which are the properties (that can be said) of and the relations (that can hold) between ontological concepts. The following are examples that illustrate the difference between logical and ontological concepts:

(7) R_1: $\text{OLD}(x :: \mathsf{Entity})$
R_2: $\text{HEAVY}(x :: \mathsf{Physical})$
R_3: $\text{HUNGRY}(x :: \mathsf{Living})$
R_4: $\text{ARTICULATE}(x :: \mathsf{Human})$
R_5: $\text{MAKE}(x :: \mathsf{Human}, y :: \mathsf{Artifact})$
R_6: $\text{MANUFACTURE}(x :: \mathsf{Human}, y :: \mathsf{Instrument})$
R_7: $\text{RIDE}(x :: \mathsf{Human}, y :: \mathsf{Vehicle})$
R_8: $\text{DRIVE}(x :: \mathsf{Human}, y :: \mathsf{Car})$

The predicates in (7) are supposed to reflect the fact that in ordinary spoken language we can say OLD of any Entity; that we say HEAVY of objects that are of type Physical; that HUNGRY is said of objects that are of type Living; that ARTICULATE is said of objects that

[4] As Hacking (2001) suggests, one can think of a type such as Cat to be the kind of object that is an answer to a question such as 'What-is-it?' Thus the distinction between types and predicates might be related to the analytic/synthetic distinction, where the truth of a type judgment such as (*sheba* :: Cat) is a synthetic judgment the truth of which is determined by virtue of what we know about the world; while the truth of the judgment WILD(*sheba* :: Cat) is determined by virtue of the meaning of WILD; i.e., by what we take WILD to mean. In this regard, and while current data-driven approaches to natural language are, in our opinion, totally misguided, type judgments like (x :: Cat) probably do belong to this quantitative level in that the truth of a type judgment such as (x :: Cat) could be determined by these pattern recognition systems, including judgments of abstract types; for example, whether a certain event is a (i.e. 'looks like' a) DANCING event.

must be of type Human; that MAKE is a relation that can hold between a Human and an Artifact; that MANUFACTURE is a relation that can hold between a Human and an Instrument, etc. Note that the type assignments in (7) implicitly define a type hierarchy as that shown in figure 1 below. Consequently, and although not explicitly stated in (7), in ordinary spoken language one can always attribute the property HEAVY to an object of type Car since Car $\sqsubseteq$ Vehicle $\sqsubseteq$ Physical[5]. In addition to logical and ontological concepts, there are also proper nouns, which are the names of objects that could be of any type. A proper noun, such as *Sheba*, is interpreted as follows $[\![Sheba]\!] \Rightarrow \lambda P[(\exists^1 Sheba :: \mathsf{Thing})(P(x))]$.

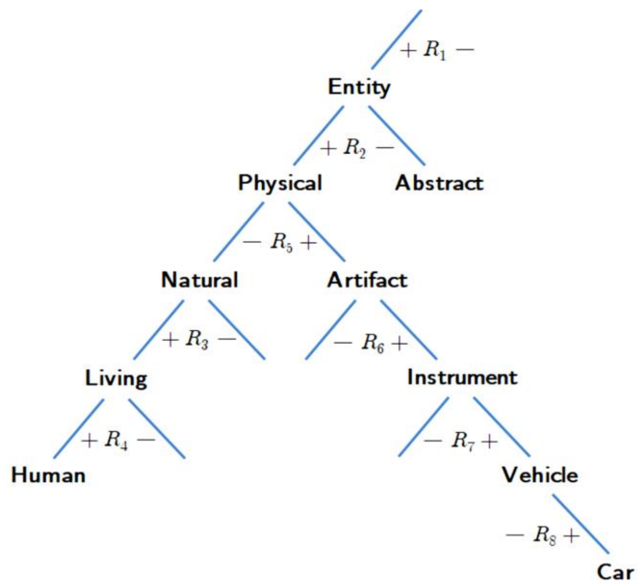

Figure 1. The type hierarchy implied by (7)

A point worth mentioning at this early juncture is that besides the embedding of 'commonsense' constraints in our predicates, what implicitly gets defined by applying Sommers' predicability test, as given by (7), is the implicit determination of 'saliency'. For example, and while it makes sense in our everyday discourse to speak of Human objects that MAKE, RIDE and DRIVE objects of type Car, DRIVE is a more salient relation between a Human and a Car since a Human rides a car as a Vehicle, and makes a car as an Artifact, but drives a car explicitly as a Car (see figure 2). We will discuss this in more detail below.

3.2 Type Unification

Let us now start our 'compositional' semantics. Consider the interpretation of *Sheba is a thief* where we assume THIEF is a property that is ordinarily said of objects that must be of type Human, that is THIEF$(x :: \mathsf{Human})$:

(8) $[\![$*Sheba is a thief*$]\!]$
$\Rightarrow (\exists^1 Sheba :: \mathsf{Thing})(\text{THIEF}(Sheba :: \mathsf{Human}))$

[5] It should be noted here that the expressions in (7) are assumed to refer to a specific sense of each predicate. In general, however, the type assignment is a set of possible types where a single type is eventually left after lexical disambiguation. This will be discussed in more detail below.

Thus, *Sheba is a thief* is interpreted as: there is a unique object named *Sheba* (initially assumed to be of type Thing), such that the property THIEF is true of *Sheba*[6].

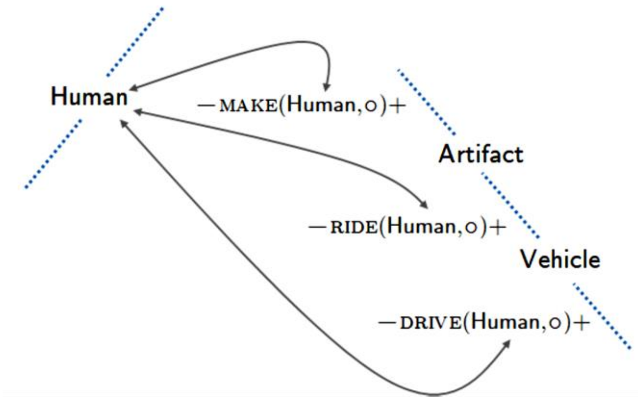

Figure 2. Salient relations implied by the ontological types and their properties.

Note now that in (8) *Sheba* is associated with more than one type in a single scope. In these situations a type unification must occur, where a type unification $(\mathsf{S} \bullet \mathsf{T})$ between S and T and where $Q \in \{\forall, \exists\}$, is defined for now as follows:

(9) $$Q(x :: (\mathsf{S} \bullet \mathsf{T}))(P(x)) \equiv \begin{cases} Q(x :: \mathsf{S})(P(x)), & \textit{if } (\mathsf{S} \sqsubseteq \mathsf{T}) \\ Q(x :: \mathsf{T})(P(x)), & \textit{if } (\mathsf{T} \sqsubseteq \mathsf{S}) \\ Q(x :: \mathsf{S})\,Q(x :: \mathsf{T})(P(x) \wedge \mathbf{R}(x,y)), & \textit{if } (\exists \mathbf{R})(\mathbf{R} = \textit{msr}(\mathsf{S},\mathsf{T})) \\ \bot, & \textit{otherwise} \end{cases}$$

where $\textit{msr}(\mathsf{S},\mathsf{T})$ stands for the most salient relation between objects of type S and objects of type T. That is, in situations where there is no subsumption relation between S and T the type unification results in keeping the variables of both types and in introducing some salient relation between the two types (we will discuss these situations below). Going to back to (8), the type unification in this case is actually quite simple, since $(\mathsf{Human} \sqsubseteq \mathsf{Thing})$:

(10) $[\![\textit{Sheba is a thief}]\!]$
$$\Rightarrow (\exists^1 \textit{Sheba} :: \mathsf{Thing})(\text{THIEF}(\textit{Sheba} :: \mathsf{Human}))$$
$$\Rightarrow (\exists^1 \textit{Sheba} :: (\mathsf{Human} \bullet \mathsf{Thing}))(\text{THIEF}(\textit{Sheba}))$$
$$\Rightarrow (\exists^1 \textit{Sheba} :: \mathsf{Human})(\text{THIEF}(\textit{Sheba}))$$

In the final analysis, therefore, *Sheba is a thief* is interpreted as follows: there is a unique object named *Sheba*, an object that eventually came out to be of type Human, such that THIEF is true of *Sheba*. Note the clear distinction between ontological concepts (e.g., Human), which Cocchiarella (2001) calls first-intension concepts, and logical (or second-intension) concepts, such as $\text{THIEF}(x :: \mathsf{Human})$. In accordance with Quine's famous slogan ("to be is to be the value of a variable"), what (10) says is that what ontologically exist are

[6] For simplicity, we are ignoring for now some intermediate steps in the translation, especially as it relates to the copula 'is' which plays an important part in determining the correct predication (more on this below).

objects of type Human, and not thieves, and THIEF is an accidental (and temporal, etc.) property that we came to use to talk of certain objects of type Human. Furthermore, it is assumed that a logical concept such as THIEF is defined by a logical expression such as $(\forall x :: \mathsf{Human})(\text{THIEF}(x) \equiv \phi)$, where the exact nature of ϕ might very well be susceptible to temporal, cultural, and other contextual factors, depending on what, at a certain point in time, a certain community considers a THIEF to be.

It should also be noted that a first-intension such as $(x :: \mathsf{Human})$ as well as a second-intension such as $\text{ARTICULATE}(x :: \mathsf{Human})$ are both 'judgments' where the former is a type judgment and the latter is a value judgment. As such, in the interpretation of 'John is articulate' the first-intension $(x :: \mathsf{Human})$ must precede the second-intension $\text{ARTICULATE}(x :: \mathsf{Human})$, unlike the 'in-parallel' evaluation $\text{HUMAN}(John) \wedge \text{ARTICULATE}(John)$ we usually get in standard first-order logic as. Another way of stating this is that for a judgment such as $\text{ARTICULATE}(x :: \mathsf{Human})$ to be made, the judgment $(x :: \mathsf{Human})$ must first be made. The way we see it, therefore, is that type judgments are the first level in the entire semantic structure, as suggested in figure 3 below.

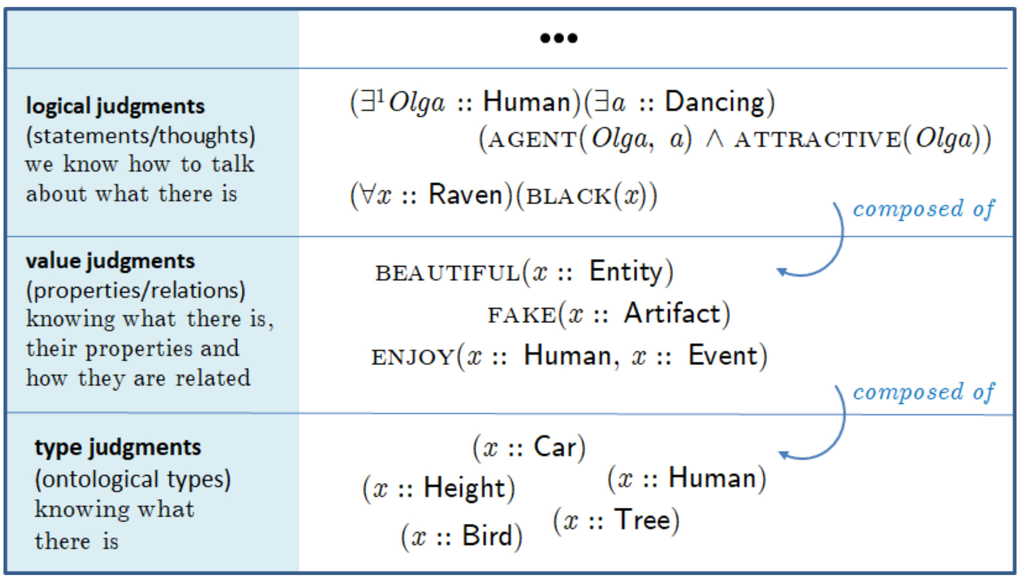

Figure 3. Levels of semantic processing.

3.3 More on Type Unification

Consider the following (initial) interpretation of *Sara owns a black cat*, where we assume $\text{BLACK}(x :: \mathsf{Physical})$ and $\text{OWN}(x :: \mathsf{Human},\ y :: \mathsf{Entity})$ – that is, we assume that BLACK can be said of objects of type Physical, and that the OWN relationship holds between objects of type Human and objects of type Entity:

(11) $[\![\textit{Sara owns a black cat}]\!]$
$\Rightarrow (\exists^1 Sara :: \mathsf{Thing})(\exists c :: \mathsf{Cat})(\text{BLACK}(c :: \mathsf{Physical})$
$\wedge\ \text{OWN}(Sara :: \mathsf{Human},\ c :: \mathsf{Entity}))$

Thus *Sara owns a black cat* is initially interpreted as follows: there is a unique Thing named *Sara*, and some object c of type Cat, such that c is BLACK (and thus in this context it must be of type Physical), and *Sara* owns c, where in this context *Sara* must be object of type Human and c an object of type Entity. Depending on the context they are mentioned in, therefore, *Sara* and c are assigned different types: initially considered to be a Thing, *Sara* is then considered to be an object of type Human (when the agent of an 'owning' relationship). The object c, on other hand, is assigned three types in a single scope: introduced as an object of type Cat, it is then considered to be an object of type Physical (when predicated by BLACK) and as an Entity (when the object of an 'owning' relation). The type unifications that must occur in this situation are the following (where '$\rightarrow$' means 'unifies to'):

$(Sara :: (\mathsf{Thing} \bullet \mathsf{Human}))$
$\rightarrow (Sara :: \mathsf{Human})$

$(c :: ((\mathsf{Physical} \bullet \mathsf{Entity}) \bullet \mathsf{Cat}))$
$\rightarrow (c :: (\mathsf{Physical} \bullet \mathsf{Cat}))$
$\rightarrow (c :: \mathsf{Cat})$

The final interpretation of *Sara owns a black cat* is finally given by the following:

(12) $[\![Sara\ owns\ a\ black\ cat]\!]$
$\Rightarrow (\exists^1 Sara :: \mathsf{Human})(\exists c :: \mathsf{Cat})(\textsc{black}(c) \wedge \textsc{own}(Sara, c))$

That is, there is unique object named *Sara*, which is of type Human, and some object c of type Cat, where c is BLACK and *Sara* owns c.

3.4 Type Unification and Abstract Objects

As discussed above, logical concepts such as TEACHER, THIEF, etc. are assumed to be defined by some logical expression. A plausible (although admittedly simplistic) definition for a logical concept such as DANCER could for example be given by (13).

(13) $(\forall x :: \mathsf{Human})(\textsc{dancer}(x) \equiv (\exists a :: \mathsf{Dancing})(\textsc{agent}(a,x))$

That is, any x (that must be of type Human) is a dancer *iff* x is the agent of some Dancing (which is a subtype Activity). Let us now consider the interpretation of *Olga is a beautiful dancer* where we assume BEAUTIFUL(a :: Entity) – i.e., 'beautiful' can be said of any Entity:

(14) $[\![Olga\ is\ a\ beautiful\ dancer]\!]$
$\Rightarrow (\exists^1 Olga :: \mathsf{Thing})(\exists a :: \mathsf{Dancing})$
$(\textsc{agent}(a :: \mathsf{Activity},\ Olga :: \mathsf{Human})$
$\wedge\ (\textsc{beautiful}(Olga :: \mathsf{Entity}) \vee \textsc{beautiful}(a :: \mathsf{Entity})))$

Thus, *Olga is a beautiful dancer* is initially translated as follows: there is a unique Thing named *Olga*, and some Dancing a, where *Olga* is the agent of this dancing, which must be an Activity (and as the agent, *Olga* must be of type Human), and where either *Olga* is BEAUTIFUL or her Dancing (or of course, both). Note now that in the same scope *Olga* and the dancing activity a are assigned three types, triggering the following type unifications:

$(Olga :: ((\mathsf{Thing} \bullet \mathsf{Entity}) \bullet \mathsf{Human}))$
$\rightarrow (Olga :: (\mathsf{Entity} \bullet \mathsf{Human}))$
$\rightarrow (Olga :: \mathsf{Human})$

$(a :: ((\mathsf{Dancing} \bullet \mathsf{Activity}) \bullet \mathsf{Entity}))$
$\rightarrow (a :: (\mathsf{Dancing} \bullet \mathsf{Entity}))$
$\rightarrow (a :: \mathsf{Dancing})$

Concerning the disjunction term in (14), representing the ambiguity in nominal modification, we now have the following $\textsc{beautiful}(Olga :: \mathsf{Human}) \vee \textsc{beautiful}(a :: \mathsf{Dancing}))$. Since both terms in the disjunction are acceptable, the final translation is the one given in (15) that admits an ambiguity in nominal modification – i.e., the possibility of 'beautiful' describing *Olga* or her dancing (or both).

(15) $[\![$*Olga is a beautiful dancer*$]\!]$
$\Rightarrow (\exists^1 Olga :: \mathsf{Human})(\exists a :: \mathsf{Dancing})$
$(\textsc{agent}(a, Olga) \wedge (\textsc{beautiful}(Olga) \vee \textsc{beautiful}(a)))$

Unlike the situation in (15), however, the relevant type unifications should remove the ambiguities in (16) and (17),

(16) *Olga is an experienced dancer*
(17) *Olga is a recreational dancer*

where it is clear that *experienced* is describing *Olga* in the former and *recreational* is describing *Olga*'s dancing in the latter (that is, in (16) it is *Olga* and not her dancing that is 'experienced', and in (17) it is not *Olga*, but her dancing that is 'recreational'). The only term we need to consider here is the term involving the disjunction representing the ambiguity in nominal modification. Here are the type unifications in the case of (16):

$\textsc{experienced}(Olga :: (\mathsf{Human} \bullet \mathsf{Human})) \vee \textsc{experienced}(a :: (\mathsf{Dancing} \bullet \mathsf{Human}))$
$\rightarrow \textsc{experienced}(Olga :: \mathsf{Human}) \vee \textsc{experienced}(a :: \bot)$
$\rightarrow \textsc{experienced}(Olga :: \mathsf{Human}) \vee \bot$
$\rightarrow \textsc{experienced}(Olga :: \mathsf{Human})$

The type unification admitting an 'experienced dancing' fails here, leaving 'experienced' to unambiguously modify *Olga*. In (17), however, we have the following:

$\textsc{recreational}(Olga :: (\mathsf{Human} \bullet \mathsf{Dancing})) \vee \textsc{recreational}(a :: (\mathsf{Dancing} \bullet \mathsf{Dancing}))$
$\rightarrow \textsc{recreational}(Olga :: \bot) \vee \textsc{recreational}(a :: \mathsf{Dancing})$
$\rightarrow \bot \vee \textsc{recreational}(a :: \mathsf{Dancing})$
$\rightarrow \textsc{recreational}(a :: \mathsf{Dancing})$

Note that in this case the type unification removes 'recreational *Olga*' leaving 'recreational' to unambiguously modify *Olga*'s dancing. One might at this point question why the type unification $(\mathsf{Dancing} \bullet \mathsf{Human})$ in (16) and (17) was considered a failure (resulting in $\bot$), although the definition of type unification given in (9) suggests that in the absence of a subsumption relation between two types S and T, an attempt is first made is to pick-up the 'most salient relation' (*msr*) between the two ontological types. The answer is that looking for an *msr* occurs when all else fails, while this is not the case in (16) and (17) where the local context provided a successful type unification and thus looking elsewhere to 'make sense' of what is being said is not needed!

4. A Short Detour

Before we turn to more linguistic phenomena that proved to be challenging in logical semantics, in this section we show that the machinery we have introduced thus far provides a very plausible explanation for two longstanding semantic puzzles.

4.1 Adjective-ordering Restrictions

4.1.1 An Innate Ontological Structure?

A longstanding subject of debate in linguistics is the phenomenon of adjective-ordering restrictions (AORs), which is concerned with the apparent preferred adjective orderings we tend to have when multiple adjectives are used in a sequence. For example, it is generally agreed that the ordering of adjectives in (18a) is preferred to that in (18b).

(18) a. *Jon bought a beautiful red car*
b. *Jon bought a red beautiful car*

What makes this an interesting phenomenon from a cognitive science point of view is the apparent universality and cross-linguistic nature of AORs, as well as the fact that children seem to effortlessly make these preferences without ever being instructed on what the most 'natural' ordering is. Various studies have suggested that these ordering preferences might be a function of some syntactic and semantic classes of adjectives (e.g., Cinque 1994; Larson, 1998; and Vendler, 1968), nouniness – how close is the adjective to a noun, or temporariness (how much does the adjective encode a temporary property) (e.g., Teodorescu, 2006). Despite all these studies, the debate as to what might explain these ordering preferences, and whether the ordering preferences might reflect a much deeper cognitive phenomenon, have not yet been settled.

What we like to suggest, however, is that adjective-ordering restrictions (that are indeed systematic), are related to type casting in a strongly-typed ontological structure. For example, the adjective orderings in (18a) and (18b) require the following type unifications:

(19) a. $\textsc{beutiful}(\textsc{red}(x :: \mathsf{Physical}) :: \mathsf{Entity})$
b. $\textsc{red}(\textsc{beutiful}(x :: \mathsf{Entity}) :: \mathsf{Physical})$

Note that the type unification in (19a) requires a type casting from Physical to Entity, while the one in (19b) requires a type casting in the reverse order. The type casting in (19b) is blocked, however, since (as it is well known in type theory of polymorphic programming languages) one can always perform type casting upwards, but not downwards (one can always view a Car as an Entity, Physical, etc., but the opposite is not true (see Figure 4)[7].

What is interesting to contemplate here is this: if the adjective-ordering restriction phenomenon (which seems to be universally and cross-linguistically valid) turns out to be explainable by type casting and type unification in a strongly-type ontological structure, it would suggest that the ontological structure we envision might have some innate underpinnings. Incidentally, recent psycholinguistic research (see Scontras *et. al.*, 2017) seems to suggest that this might be the case.

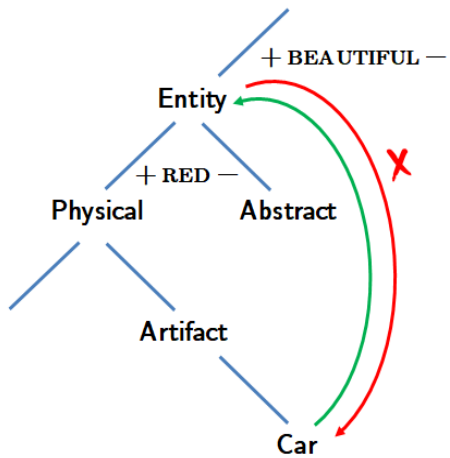

Figure 4. Adjective-Ordering Restrictions and Type Casting

4.1.2 Interaction between Type Casting and Type Unification

Recall the interpretation of *Olga is a beautiful dancer* in (15), where the final interpretation admitted the ambiguity in nominal modification where all type unifications succeeded, allowing 'beautiful' to remain ambiguous in that it could be modifying *Olga* or her dancing. Let us now consider the interpretation of *Olga is a beautiful tall dancer*, where we assume $\text{TALL}(x :: \mathsf{Physical})$:

(20) $[\![\textit{Olga is a beautiful tall dancer}]\!]$

$$\Rightarrow (\exists^1 Olga :: \mathsf{Thing})(\exists a :: \mathsf{Dancing})$$
$$(\text{AGENT}(a :: \mathsf{Activity}, Olga :: \mathsf{Human})$$
$$\wedge (\text{BEAUTIFUL}(\text{TALL}(Olga :: \mathsf{Physical}) :: \mathsf{Entity})$$
$$\vee \text{BEAUTIFUL}(\text{TALL}(a :: \mathsf{Physical}) :: \mathsf{Entity})))$$

The type unifications concerning required here are the following:

$(Olga :: ((\mathsf{Thing} \bullet \mathsf{Physical}) \bullet \mathsf{Human}))$
$\rightarrow (Olga :: (\mathsf{Physical} \bullet \mathsf{Human}))$
$\rightarrow (Olga :: \mathsf{Human})$

[7] Technically, we can always generalize (cast up) by ignoring specific details; casting down (assuming unknown details), is however undecidable.

$(a :: ((\mathsf{Dancing} \bullet \mathsf{Activity}) :: \mathsf{Physical}))$
$\rightarrow (a :: (\mathsf{Dancing} \bullet \mathsf{Physical}))$
$\rightarrow (a :: \bot)$
$\rightarrow \bot$

resulting in (21):

(21) $[\![\textit{Olga is a beautiful tall dancer}]\!]$
$\Rightarrow (\exists^1 Olga :: \mathsf{Human})(\exists a :: \mathsf{Dancing})$
$(\textsc{agent}(a,\ Olga)$
$\wedge\ (\textsc{beautiful}(\textsc{tall}(Olga :: \mathsf{Human})) \vee \textsc{beautiful}(\textsc{tall}(a :: \bot) :: \mathsf{Entity})))$
$\Rightarrow (\exists^1 Olga :: \mathsf{Human})(\exists a :: \mathsf{Dancing})$
$(\textsc{agent}(a,\ Olga) \wedge (\textsc{beautiful}(\textsc{tall}(Olga)) \vee \bot))$
$\Rightarrow (\exists^1 Olga :: \mathsf{Human})(\exists a :: \mathsf{Dancing})$
$(\textsc{agent}(a,\ Olga) \wedge \textsc{beautiful}(\textsc{tall}(Olga)))$

Unlike the situation in (15), where ‘beautiful’ could be describing *Olga* or her dancing, the situation in (21) is quite different due to the adjective ‘tall’ that forced ‘beautiful’ to be describing a physical object, an thus *Olga*, and not her dancing.

We leave it to the reader to work out why (and how) *Olga is a beautiful tall dancer* sounds fine, while *Olga is a tall beautiful dancer* sounds awkward (hint: in the latter, the type castings and type unifications reduce the entire disjunction to ⊥!)

4.2 What ‘Paradox of the Ravens’?

Introduced in the 1940’s by Carl Gustav Hempel, then a student of Carnap, the Paradox of the Ravens – or Hempel’s Paradox, or the Paradox of Confirmation (Hempel, 1945) has continued to occupy logicians, statisticians, and philosophers of science to this day. The paradox arises when one considers what constitutes an evidence for a statement (hypothesis) and it can be described by the following:

(22) H1: *All ravens are black*
H2: *All non-black things are not ravens*

The hypothesis H1 is logically equivalent to the hypothesis H2, as is typically shown in the following translation in standard first-order predicate logic:

(23) H1: $(\forall x)(\textsc{raven}(\mathrm{x}) \supset \textsc{black}(\mathrm{x}))$
H2: $(\forall x)(\neg\textsc{black}(\mathrm{x}) \supset \neg\textsc{raven}(\mathrm{x}))$

The assumed ‘paradox’ is now due to the irreconcilability of the following, separately valid claims: (*i*) any observation of an instance that satisfies H1 is said to confirm H1 (to some degree); and (*ii*) any observation that confirms a hypothesis H1 must confirm (and to the same degree) a hypothesis that is logically equivalent to it; and (*iii*) the observation of any non-black thing that is also not a raven confirms H2, and thus, it conforms the logically equivalent hypothesis H1. This, however, leaves us with the unpleasant conclusion that observing a white shoe, a blue shirt, or, for that matter, any non-black and non-raven object currently in your sight, confirming 'all ravens are black'. Clearly, this cannot be accepted, since the observation of a red herring, say, has no bearing on the hypothesis that 'all ravens are black'.

Many solutions have been proposed to the Paradox of the Ravens that range from accepting the paradox (that observing red apples and other non-black non-ravens does confirm the hypothesis 'All ravens are black' – since the observation *does not disconfirm* it), to proposals in the Bayesian tradition that try to measure the 'degree' of confirmation. Concerning the latter, Bayesian proposals essentially amount to proposing that observing a red apple, for example, does confirm the hypothesis 'All ravens are black' but it does so very minimally, and certainly much less than the observation of a black raven (see Vranas, 2004). Clearly, however, this is not a satisfactory solution since observing a red flower should not contribute at all to the confirmation of 'All ravens are black'. Worse yet, it turns out that in the Bayesian analysis the observation of black but non-raven objects actually negatively confirms the hypothesis that 'All ravens are black', which is also problematic. An even more serious flaw in the Bayesian analysis is this: if observing a red apple confirms the hypothesis that 'All ravens are black' (no matter how minutely), then it also minimally, but equally, confirms the hypothesis that 'All ravens are green'. Thus we have the same observation that confirms two mutually exclusive hypotheses, which clearly is unacceptable (see Maher, 1999 for a more detailed discussion).

What we suggest here is that the purported 'paradox' is not a logical one, but a representational one. In particular, we believe that the problem lies in the standard first-order predicate logic formulation of the hypothesis and its contrapositive. Let us suggest embedding ontological types in the standard first-order logic representation, where we are assuming $\text{BLACK}(x :: \mathsf{Physical})$, i.e., BLACK is a property that is ordinarily said of objects of type $\mathsf{Physical}$. We will start first with H1, namely the hypothesis *All ravens are black*:

(24)
1. $[\![All]\!] \Rightarrow \lambda P \lambda Q[(\forall x :: \mathsf{Thing})(P(x) \supset Q(x))]$
2. $[\![ravens]\!] \Rightarrow \lambda x[(x :: \mathsf{Raven})]$
3. $[\![All\ ravens]\!]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Thing})((x :: \mathsf{Raven}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: (\mathsf{Thing} \bullet \mathsf{Raven}))((x :: \mathsf{Raven}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Raven})((x :: \mathsf{Raven}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Raven})(\mathbf{true} \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Raven})(Q(x))]$
4. $[\![are]\!] \Rightarrow \lambda P \lambda x[P(x :: \mathsf{Thing})]$
5. $[\![black]\!] \Rightarrow \lambda x[(\text{BLACK}(x :: \mathsf{Physical}))]$
6. $[\![are\ black]\!] \Rightarrow \lambda x[(\text{BLACK}(x :: \mathsf{Physical}))]$
7. $[\![All\ ravens\ are\ black]\!]$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\text{BLACK}(x :: \mathsf{Physical}))$
   $\Rightarrow (\forall x :: (\mathsf{Raven} \bullet \mathsf{Physical}))(\text{BLACK}(x))$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\text{BLACK}(x))$

Note the crucial step (24.3), where unifying the term $(x :: \mathsf{Raven})$ with the variable introduced by the quantifier reduces the term $(x :: \mathsf{Raven})$ to **true**, and thus the meaning of *All ravens* to $\lambda Q[(\forall x :: \mathsf{Raven})(Q(x))]$. In (24.7), the term $\text{BLACK}(x :: \mathsf{Physical})$ forces a type unification ($\mathsf{Raven} \bullet \mathsf{Physical}$), resulting in the final interpretation of *All ravens are black* as $(\forall x :: \mathsf{Raven})(\textsc{black}(x))$ – that is, for any object x of type Raven, $\textsc{black}$ is true of x. Let us now consider the logically equivalent hypothesis H2, namely that *All non-black things are not ravens*:

(28)
1. $[\![All]\!] \Rightarrow \lambda P \lambda Q[(\forall x :: \mathsf{Thing})(P(x) \supset Q(x))]$
2. $[\![non\text{-}black]\!] \Rightarrow \lambda x[\neg\textsc{black}(x :: \mathsf{Physical})]$
3. $[\![All\ non\text{-}black]\!]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Thing})(\neg\textsc{black}(x :: \mathsf{Physical}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: (\mathsf{Thing} \bullet \mathsf{Physical}))(\neg\textsc{black}(x :: \mathsf{Physical}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Physical})(\neg\textsc{black}(x :: \mathsf{Physical}) \supset Q(x))]$
   $\Rightarrow \lambda Q[(\forall x :: \mathsf{Physical})(\neg\textsc{black}(x) \supset Q(x))]$
4. $[\![are]\!] \Rightarrow \lambda P \lambda x[P(x :: \mathsf{Thing})]$
5. $[\![non\text{-}ravens]\!] \Rightarrow \lambda x[\neg(x :: \mathsf{Raven})]$
6. $[\![are\ non\text{-}ravens]\!] \Rightarrow \lambda x[\neg\textsc{black}(x :: \mathsf{Physical})]$
7. $[\![All\ non\text{-}black\ are\ non\text{-}ravens]\!]$
   $\Rightarrow (\forall x :: \mathsf{Physical})(\neg\textsc{black}(x) \supset \neg(x :: \mathsf{Raven}))$
   $\Rightarrow (\forall x :: (\mathsf{Raven} \bullet \mathsf{Physical}))(\neg\textsc{black}(x) \supset \neg(x :: \mathsf{Raven}))$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\neg\textsc{black}(x) \supset \neg(x :: \mathsf{Raven}))$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\neg\textsc{black}(x) \supset \neg\mathbf{true})$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\mathbf{false} \vee \textsc{black}(x))$
   $\Rightarrow (\forall x :: \mathsf{Raven})(\textsc{black}(x))$

Note that in this situation the widest scope variable x remains to be of type $\mathsf{Physical}$ until the last step (28.7) where the unification ($\mathsf{Raven} \bullet \mathsf{Physical}$) casts x to a specific $\mathsf{Physical}$ object, namely Raven. What matters for us here is that in a logical semantics embedded with types from a strongly-typed ontology both *All ravens are black* and its logically equivalent *All non-black things are not ravens* turn out to be equivalent not only in content, but also in form, where the logical formulation of both is $(\forall x :: \mathsf{Raven})(\textsc{black}(x))$, and thus there is no 'paradox' of the ravens since H1 and H2 are now confirmed (and disconfirmed) by the same observations!

5. Where Logical Semantics Can (Should?) Go

In this section we show how the embedding of ontological concepts in our predicates can help us tackle some well-known challenges in the semantics of natural language.

5.1 Lexical Disambiguation

For now we have been assuming single type assignments to the variables of our predicates, e.g. $\textsc{black}(x :: \mathsf{Physical})$, $\textsc{beautiful}(x :: \mathsf{Entity})$, etc. as it was implicitly assumed that the predicate in question has been disambiguated – that is, that a specific meaning of the predicate has been selected. We will continue to do so where the context is clear, although we will show hear how lexical disambiguation itself is conducted in our system, and that requires that we initially consider, for some terms, a set of type assignments.

Let us consider the interpretation of the sentence in (26), where we will concentrate on two senses of 'party' – i.e., we will assume that 'party' belongs to (at least) two branches in our ontological structure, as shown in Figure 5.

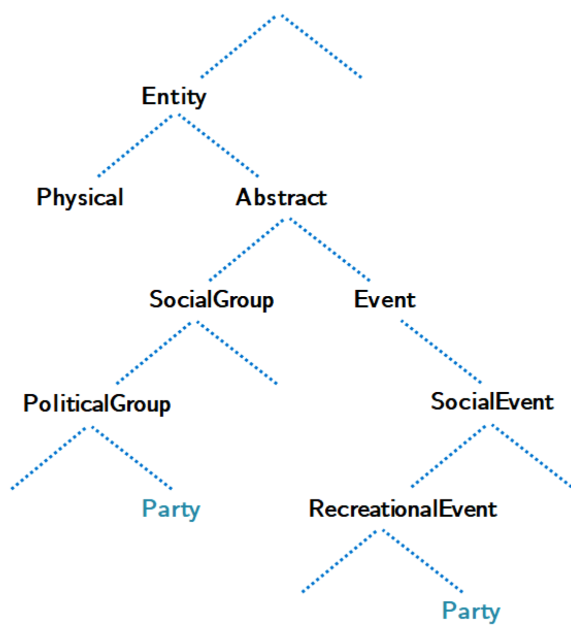

Figure 5. Lexical ambiguity as multiple ontological types

(29) ⟦*Jon cancelled the party*⟧
$\Rightarrow (\exists^1 Jon :: \mathsf{Thing})(\exists a :: \mathsf{Activity})(\exists^1 p :: \{\mathsf{PoliticalGroup}, \mathsf{RecreationalEvent}\})$
$(\textsc{cancellation}(a) \wedge \textsc{subject}(a, Jon :: \mathsf{Human})) \wedge \textsc{object}(a, p :: \mathsf{Event}))$

Thus, initially, the types associated with 'the party' p is a set of all possible types (again, for simplicity we assumed that 'cancelled' has been disambiguated where it was determined that it is some activity the object of which is an object of type Event). The type unifications concerning *Jon* are straightforward. The type unifications that must occur for p are now a set of n pairs of type unifications, where n is the number of possible meanings of p:

$(p :: \{(\mathsf{Event} \bullet \mathsf{PoliticalGroup}),(\mathsf{Event} \bullet \mathsf{RecreationalEvent})\})$
$\rightarrow (p :: \{\bot,\mathsf{RecreationalEvent}\})$
$\rightarrow (p :: \{\mathsf{RecreationalEvent}\})$

Thus, the initial set of types is reduced to a singleton and the 'party' that John seems to have cancelled is the 'recreational event' meaning of 'party'. Note that if 'cancelled' in (29) was replaced by 'assisted', where it is assumed that the object of 'assistance' can be a human or a social group, then the correct meaning of 'party' will also be selected, namely the meaning of the social (political) group. On the other hand, if 'cancelled' where to be replaced by 'promoted' then we would have a genuinely ambiguous statement, since one can 'promote' a political group, as well as a recreational event, as illustrated by (30).

(30) $[\![$*Jon promoted the party*$]\!]$
$\Rightarrow (\exists^1 Jon :: \mathsf{Thing})(\exists a :: \mathsf{Activity})(\exists^1 p :: \{\mathsf{PoliticalGroup}, \mathsf{RecreationalEvent}\})$
$(\textsc{promotion}(a) \wedge \textsc{subject}(a, Jon :: \mathsf{Human}))$
$\wedge\ \textsc{object}(a, p :: \mathsf{Entity}))$

Assuming that any Entity can be promoted, and in the absence of any additional information (and thus additional type constraints), both meanings of 'party' remain to be equally plausible.

5.2 Metonymy, Copredication and Salient Meanings

Consider the sentence in (31), where two senses of 'book' are assumed to be used in the same context, the informational content sense of book (when being read) and the physical object sense (when being burned):

(31) *Jon read the book and then he burned it.*

In Asher and Pustejovsky (2005) it is argued that this is an example of what they term copredication; where incompatible predicates are applied to the same type of object. It is argued that in (31), for example, 'book' must have what is called a dot type, which is a structured object that in a sense carries the 'informational content' sense (which is referenced when it is being read) as well as the 'physical object' sense (which is referenced when it is being burned). Elaborate machinery is then introduced to 'pick out' the right sense in the right context, and all in a well-typed compositional logic. But this approach presupposes that one can enumerate, *a priori*, all possible uses of the word 'book' in ordinary language (recall also example (7) and the relevant discussion in section 2). More to the point, we believe that is what termed 'copredication' is not different, in essence, from what is known as metonymy, as in both cases we are trying to either 'pick-up' (*i*) a hidden essence/sense of some object; or (*ii*) some hidden (i.e., unstated) relation between some objects in the discourse under consideration. To illustrate this point further, let us first consider the interpretation of (32).

(32) $[\![$*Jon bought and studied Das Kapital*$]\!]$
$\Rightarrow (\exists^1 Jon :: \mathsf{Thing})(\exists^1 DasKapital :: \mathsf{Book})(\exists a_1 :: \mathsf{Activity})(\exists a_2 :: \mathsf{Activity})$
$(\textsc{studying}(a_1) \wedge \textsc{subject}(a, Jon :: \mathsf{Human})$
$\wedge\ \textsc{object}(a, DasKapital :: \mathsf{InformationalMaterial}) \wedge$
$\textsc{buying}(a_2) \wedge \textsc{subject}(a, Jon :: \mathsf{Human})$
$\wedge\ \textsc{object}(a, DasKapital :: \mathsf{Physical}))$

That is, Jon bought a book that he also studied, where the object of a buying activity must be a Physical object, and what a Human can study is some InformationalMaterial. Again the type unifications involving *Jon* are straightforward. Similarly, the type unifications involving the Physical book *DasKapital* are also very simple:

$(DasKapital :: (\mathsf{Physical} \bullet \mathsf{Book})) \rightarrow (DasKapital :: \mathsf{Book})$

However, the only alternative for the object of the STUDYING activity is for Book and InformationalMaterial to somehow unify, and this happens when the most salient relation between Book and InformationalMaterial is picked-up, introducing in the process a new variable *infoc* that is related to a book as HASCONTENT(*DasKapital*, *infoc*). The final translation is thus that given by (33).

(33) ⟦*Jon bought and studied Das Kapital*⟧
$\Rightarrow (\exists^1 Jon :: \mathsf{Human})(\exists^1 DasKapital :: \mathsf{Book})$
$(\exists a_1 :: \mathsf{Activity})(\exists a_2 :: \mathsf{Activity})(\exists infoc :: \mathsf{InformationalMaterial})$
$(\text{STUDYING}(a_1) \wedge \text{SUBJECT}(a, Jon) \wedge \text{OBJECT}(a, DasKapital) \wedge$
$\text{BUYING}(a_2) \wedge \text{SUBJECT}(a, Jon) \wedge \text{OBJECT}(a, DasKapital) \wedge$
$\text{HASCONTENT}\ (DasKapital, infoc))$

That is, *Jon bought and studied Das Kapital* describes a situation where there is a unique object named *Jon*, an object of type Human, and some Book named *DasKapital* (that *Jon* bought), and where *DasKapital* has the InformationalMaterial *infoc* (that *Jon* studied).

It is important to note at this stage that hidden (and implicitly assumed) information is either obtained by straight forward type unification or, when all attempts fail, by picking up some salient property or relation between the objects in the discourse. On the other hand, unwanted meanings (as in lexical disambiguation, or removing some ambiguities related to nominal modification) are obtained when certain type unifications fail.

Another important point that we like to make here is *copredication*, the name given for the phenomenon that is exemplified by (34), which is not much different from what is known by metonymy. Consider for example the following:

(34) ⟦*The omelet wants a beer*⟧
$\Rightarrow (\exists^1 oml :: \mathsf{Omelet})(\exists b :: \mathsf{Beer})(\exists e :: \mathsf{Event})$
$(\text{WANTING}(e) \wedge \text{SUBJECT}(a, oml :: \mathsf{Human}) \wedge \text{OBJECT}(a, b :: \mathsf{Thing}))$

In this case, resolving the situation of 'what is wanted' is quite simple: the object of wanting is an object of Thing, and more specifically it is Beer, which works very well. However, it is the subject of the wanting that seems to be the problem: 'want' expects a Human subject but we found an Omelet. Clearly, these two types must somehow be reconciled. Since no subsumption relation exists between these types, the only way they can be reconciled is by finding some salient relation between them. As it turns out, there is a salient relationship between a Human and Food (a supertype of Omelet), namely the EAT relation, that will necessarily introduce an (implicit) object of type Human. Thus,

(35) ⟦*The omelet wants a beer*⟧
$\Rightarrow (\exists^1 oml :: \mathsf{Omelet})(\exists b :: \mathsf{Beer})(\exists e :: \mathsf{Event})(\exists a :: \mathsf{Activity})(\exists p :: \mathsf{Human})$
$(\text{EATING}(a) \wedge \text{SUBJECT}(a, p) \wedge \text{OBJECT}(a, oml)$
$\text{WANTING}(e) \wedge \text{SUBJECT}(a, p) \wedge \text{OBJECT}(a, b))$

That is, there is some eating and some wanting that is going on, and the object of the eating is an omelet, and the subject of the eating wants a beer.

5.3 Ontological Types and the Copula

It is widely accepted in formal semantics that there are two senses for the coupla 'is' in natural language. The two senses, generally attributed to Frege, are the 'is of identity', exemplified by (36) and the 'is of predication', exemplified by (37) (see Mendelsohn, 1987).

(36) *Billy the Kid is William H. Bonney.*
(37) *Billy the Kid is an outlaw.*

While (Mendelsohn, 1987) rejects the notion of two senses of 'is', and correctly so, in our opinion, his treatment of 'is' boils down to considering 'identity' as a special case of predication, where uniqueness of identity is guaranteed by some special axioms to the theory. Ironically, Mendelsohn's proposal was inspired by the work of Sommers (1969), whose ideas on types and ontology (Sommers, 1963) are very much inline with the proposal we make here. However, we believe that those proposals are lacking in that the ambiguity is not due to the flexibility of the copula 'is' but in that fact that this copula can is highly polymorphic and can therefore trigger different interpretation depending on the types of objects that are flanked on both sides. To be sure, even in the case of (proper) names, there is no 'real' identity, except in the vacuous case of 'x is x' since $7 + 8$, is not exactly 'identical' to $\sqrt{256}$, for example, as illustrated by (38).

(38) *Mary taught her little brother that* $7 + 8 = 16$.
$\not\supset$ *Mary taught her little brother that* $7 + 8 = \sqrt{256}$.

What matters to us here is to demonstrate how the copula 'is' is treated in our system. For us, 'is' is a polymorphic function that has the general template $\textsc{is}(x :: \mathsf{S},\ y :: \mathsf{T})$ where the final interpretation is a function of the types S and T and their type unification. That is, in cases where there is no subsumption relationship between S and T, the copula 'is' will introduce a salient relation between the two types. In (39) we have some common examples that involve very general and abstract types, where $\mathbf{HASprop}(x,y)$ means x has the property y, $\mathbf{INPro}(x,y)$ means x is in (or is going through) the process y, $\mathbf{DOES}(x,\ y)$ means x (often) does y, and $\mathbf{INSt}(x,\ y)$ means x is in the state y.

(39) a. *Liz is famous*
$\Rightarrow (\exists^1 Liz :: \mathsf{Human})(\exists p :: \mathsf{Property})(\textsc{fame}(p) \wedge \mathbf{HASprop}(Liz,\ p))$
b. *Jon is aging*
$\Rightarrow (\exists^1 Jon :: \mathsf{Human})(\exists p :: \mathsf{Process})(\textsc{aging}(p) \wedge \mathbf{INPro}(Liz,\ p))$
c. *Aging is inevitable*
$\Rightarrow (\forall p{::}\mathsf{Process})(\textsc{aging}(p) \supset (\exists e{::}\mathsf{Property})(\textsc{inevitability}(e) \wedge \mathbf{HASprop}(p,e)))$
d. *Fame is desirable*
$\Rightarrow (\forall f\ {::}\mathsf{Property})(\textsc{fame}(p) \supset (\exists d{::}\mathsf{Property})(\textsc{desirability}(d) \wedge \mathbf{HASprop}(f,\ d)))$
e. *Olga is a dancer*
$\Rightarrow (\exists^1 Olga :: \mathsf{Human})(\exists a :: \mathsf{Activity})(\textsc{dancing}(a) \wedge \mathbf{DOES}(Olga,\ a))$
f. *Sheba is dead*
$\Rightarrow (\exists^1 Sheba :: \mathsf{Human})(\exists s :: \mathsf{State})(\textsc{death}(s) \wedge \mathbf{INSt}(Olga,\ a))$

For example, (39a) says that *Liz* has the property of FAME, while (39c) says that the property of FAME is desirable, etc.

6. On the Nature of the Ontological Structure

Throughout this paper we have assumed the existence of some ontological structure, an ontological structure the types of which are assumed to be embedded in predicates (the properties of and the relations between objects of various types). However, a valid question that one might ask is the following: how does one arrive at this ontological structure that implicitly underlies all that we say in everyday discourse? One plausible answer is the (*seemingly* circular) suggestion that the semantic analysis of natural language should itself be used to uncover this structure. In this regard we strongly agree with Dummett (1991) who states:

> We must not try to resolve the metaphysical questions first, and then construct a meaning-theory in light of the answers. We should investigate how our language actually functions, and how we can construct a workable systematic description of how it functions; the answers to those questions will then determine the answers to the metaphysical ones.

What this suggests, and correctly so, in our opinion, is that in our effort to understand the complex and intimate relationship between ordinary language and everyday (commonsense) knowledge, one could, as Bateman (1995) has also suggested, "use language as a tool for uncovering the semiotic ontology of commonsense" since language is the only theory we have of everyday knowledge. To alleviate this seeming circularity (in assuming this ontological structure in our semantic analysis; while at the same time suggesting that semantic analysis of language should itself be used to uncover this ontological structure), we suggest performing semantic analysis from the ground up, assuming a minimal (almost a trivial and basic) ontology, building up the ontology as we go guided by the results of the semantic analysis. The advantages of this approach are: (*i*) the ontology thus constructed as a result of this process would not be invented, as is the case in most approaches to ontology (e.g., Guarino, 1995, Lenat and Guha, 1990, and Sowa, 1995), but would instead be discovered from what is in fact implicitly assumed in our use of language in everyday discourse; (*ii*) the semantics of several natural language phenomena should as a result become trivial, since the semantic analysis was itself the source of the underlying knowledge structures (in a sense, one could say that the semantics would have been done before we even started!)

Another promising technique that can be used to 'boot-up' this ontological structure is performing some corpus analysis to initially obtain sets of monadic predicates that seem to be (sensibly) used in the predication of some very general types (e.g., Artifact, Event, Physical, State, etc.) A subset relationship analysis can then be used to discover the hidden hierarchical structure. A similar approach has been suggested in Saba (2006).

7. Concluding Remarks

Most of the challenges in the semantics of natural language seem to be related to the phenomenon of the 'missing text' – that is, text is almost never explicitly stated but is implicitly assumed as 'shared' background knowledge. As Hobbs (1985) has suggested, however, challenges in the semantic analysis of natural language can become more tractable if our logical semantics was embedded with some theory of the commonsense world, at least as we talk about it in our everyday discourse.

In this paper we suggested one such method, a method that rectifies what we believe was an oversight in logical semantics, namely distinguishing between two fundamentally different types of concepts: (i) ontological concepts, that correspond to types in a strongly-typed ontology; and (ii) logical concepts, that correspond to the properties of and the relations between objects of various ontological types. One method to distinguish between the two was vaguely suggested by early work in the metaphysics of natural languages (Sommers, 1963) and is also related to some very powerful ideas put forward by Cocchiarella (2001) where he makes a strong argument for a new 'logic as language' – a logic that has ontological content where one can clearly distinguish between what he calls first-intension and second-intension concepts. We have tested our approach on a number of challenges in semantics, where we suggested clear and uniform treatment of a number of phenomena – phenomena that are usually treated by proposing often incompatible solutions.

Several linguistic phenomena, some of which have been relegated to intensionality (e.g., intensional verbs), or to reasoning with abstract objects (such as '*Jon is wise*' should follow from '*exercising is wise*' and '*Jon is exercising*'), or to compound nominals, quantifier scope ambiguity, etc. were not dealt with in this paper as that would require some minor extension to our formalism and would thus extend the paper considerably. These and a more detailed presentation of some of the subjects covered in this paper are forthcoming. Much is left to be done, refined, and restated, of course, especially as it relates to the 'discovery' of that ontological structure that seems to be implicit in everything we say in our everyday discourse.

References

1. Asher, N. 2008, A type driven theory of predication with complex types. *Fundamenta Informaticae*, **84** (2), pp. 151-183.
2. Asher, N. 2011, *Lexical Meaning in context - a web of words.* Cambridge University press.
3. Asher, N. 2015, Types, meanings and coercions in lexical semantics, *Lingua*, 157, pp. 66-82.
4. Asher, N. and Pustejovsky, J., 2012, A type composition logic for generative lexicon. *Advances in Generative Lexicon Theory.* Pustejovsky, J., P. Bouillon, H. Isahara, K. Kanzaki, and C. Lee (eds.). Springer.
5. Asher, N. and Pustejovsky, J. (2005), Word Meaning and Commonsense Metaphysics, available at the semanticsarchive.net.
6. Bateman, J. A. (1995), On the Relationship between Ontology Construction and Natural Language: A Socio-Semiotic View, *International J. of Human-Computer Studies,* **43**, pp. 929-944.
7. Cinque, G. 1994, On the Evidence for Partial N-Movement in the Romance DP. *Paths Towards Universal Grammar.* Guglielmo Cinque, et. al. (eds.), Georgetown University Press, 85-110
8. Cocchiarella, N. B. 2001, Logic and Ontology, *Axiomathes*, **12**, pp. 117-150.
9. Dummett. M. (1991), *Logical Basis of Metaphysics,* Duckworth, London.
10. Fodor, J. 1998, *Concepts: Where Cognitive Science Went Wrong*, Oxford University Press.
11. Frege, G. 1969, On concept and object, In *Translations from the Philosophical Writings of Gottlob Frege*, ed., P. Geach and M. Black, Blackwell, Oxford.

12. Givon, T. 1984, Deductive Vs. Pragmatic Processing in Natural Language, In W. Kintsch, et. a;. (Eds.), *Methods and Tactics in Cognitive Science*, 137-189, LEA Acosiates.
13. Hacking, I. 2001, Aristotelian Categories and Cognitive Domains, *Synthese* **126**: 473–515.
14. Hempel, G. 1945, Studies in the Logic of Confirmation, *Mind*, **54** (213), pp. 1-26.
15. Hobbs, J. 1985, Ontological Promiscuity, In *Proceedings of the 23rd Annual Meeting of the Assoc. for Computational Linguistics*, pp. 61-69.
16. Hobbs, J. 1985, Ontological Promiscuity, In *Proceedings of the 23rd Annual Meeting of the Assoc. for Computational Linguistics*, pp. 61-69.
17. Larson, R. K. 1998, Events and Modification in Nominals, In *Proceedings of Semantics and Linguistic Theory* (SALT) VIII, Devon Strolovitch & Aaron Lawson (eds.), pp. 145-168
18. Levesque, H. J. 2011, The Winograd Schema Challenge, In *AAAI 2011 Spring Symp. on Logical Formalizations of Commonsense Reasoning,* pp. 63-68
19. Luo, Z. 2010, Type-theoretical semantics with coercive subtyping. In *Semantics and linguistic Theory* 20 (SALT20), Vancouver.
20. Luo, Z. 2011, Contextual analysis of word meanings in type-theoretical semantics. *Logical Aspects of Computational Linguistics* (LACL'2011). LNAI 6736.
21. Luo, Z. 2012, Formal semantics in modern type theories with coercive subtyping, *Linguistics and Philosophy*, 35:491–513
22. Maher, P. 1999, Inductive Logic and the Ravens Paradox, *Philosophy of Science*, **66**, pp. 50-70.
23. Martin-Löf, P., 1984, *Intuitionistic type theory.* Napoli: Bibliopolis.
24. Mendelsohn, R. L. 1987, Frege's Two Senses of 'Is', Notre Dame Journal of Formal Logic, 28 (1), pp. 139-160.
25. Montague, R. 1974, *Formal Philosophy: Selected Papers of Richard Montague.* Edited and with an introduction by Richmond H. Thomason, Yale University Press.
26. Pustejovsky, James, 1995, *The Generative Lexicon.* MIT Press.
27. Scontras, G., Degen, J. and Goodman, N. D. 2017, Subjectivity Predicts Adjective Ordering Preferences, *Open Mind: Discoveries in Cognitive Science,* **1** (1), pp. 53-65.
28. Sommers, F. 1963, Types and Ontology, *Philosophical Review,* **72** (3), pp. 327-363.
29. Sommers, F. 1969, Do we need identity?, *The Journal of Philosophy*, vol. LXVI, pp. 499-504
30. Teodorescu, A. 2006, Adjective ordering restrictions revisited, In *Proceedings of the West Coast Conference on Formal Linguistics* (WCCFL) 25, pp. 399–407.
31. Vranas, P. B. M., 2004, Hempel's Raven Paradox: A Lacuna in the Standard Bayesian Solution, *British Journal for the Philosophy of Science*, 55, pp. 545-560.