Some Applications of Category Theory to Natural Language Interpretation

Ash Asudeh
[Joint work with Gianluca Giorgolo]

January 24, 2017

1 Setting the Linguistic Scene

- For the last five years or so, Gianluca Giorgolo and I have been exploring using monads to model certain murkier aspects of natural language meaning [Giorgolo and Asudeh, 2011, 2012a,b, 2014a,b, Asudeh and Giorgolo, 2015], following ideas that came to us through Shan [2001].
- We sometimes call the general approach *meaning enrichment* and the semantic pieces *enriched meanings*.
- In some sense, this is more a research program than a specific result, so we hope to give you a notion of some of the sorts of things that we've thought of to do with monads for natural language analysis.
- Our normal audience potentially knows the phenomena well but are new to the formal category-theoretic tools. I'm assuming that this group is in the opposite situation; i.e. that you know the formal tools far better than we do (certainly than I do), but are probably less informed about the linguistic phenomena.
- So I'll try to focus more on the latter, but then briefly sketch the category-theoretic analysis we give for each of the three phenomena.

Semantics & Pragmatics

- What do we construe these to be, in general terms?

  Semantics:
  - Meanings of linguistic expressions
  - Conventionalized
  - Truth-conditional

  Pragmatics:
  - Meanings of uses of linguistic expressions
  - Non-conventionalized
  - Non-truth-conditional

- Unfortunately, this doesn't seem to carve nature at its joints: lots of phenomena having to do with natural language meaning seem to show mixtures of *conventionality* and *truth-conditionality* (we follow Gutzmann 2015 in this way of laying things out, but the same point has been raised in many guises previously). This is shown in Table 1.
- The cells marked [+conventional,+truth-conditional] and [−conventional,−truth-conditional] are respectively "clear-cut" cases of semantics and pragmatics.

	+ truth-conditional	− truth-conditional
+ conventional	descriptive meaning	use-conditional meaning
− conventional	pragmatic enrichment	conversational implicatures

Table 1: Conventions vs. truth conditions [Gutzmann, 2015, 5]

- The other two cells (*pragmatic enrichment* and *use-conditional meaning*) are the borderlands between semantics & pragmatics, where a lot of the interesting action is.
- It is the borderlands that Gianluca and I have been concerned with.
- What is meant by *conventionality*?
  To say that a meaning is conventional is to say that it is part of a regular form–meaning mapping. Conventionality is in some sense a corollary of compositionality: if the meanings of larger expressions can be determined on the basis of the meanings of their parts (and syntax), then the parts must also have meanings. This process must ultimately bottom out in a *conventional*, i.e. regular and predictable, specification of meaning for the smallest meaningful parts (morphemes, words, constructions — whatever floats your boat). Another way to understand conventionality is as the component of meaning that is not sensitive to real-world (i.e., extra-linguistic) knowledge.
- What is meant by *truth-conditionality*?
  It is common-place to understand the meaning of a sentence as its truth-conditions — with suitable but compatible modifications to capture meanings of non-declaratives — and to understand the meanings of its parts based on how they contribute to these truth-conditions. Again, truth-conditionality is related to compositionality: it is one way of enacting the Fregean idea.
- **Descriptive meaning**

  Consider the following simple but apt examples from Gutzmann [2015, 2]:

  (1) a. A cat sleeps under the couch.
      b. A turtle sleeps under the couch.

  (2) a. The turtle sleeps under the couch.
      b. The turtle sleeps under the sofa.

  The first two sentences have different truth-conditions. Since the only difference is *cat* versus *turtle*, this must be the source. Therefore, *cat* and *turtle*, have different truth-conditional meanings and this is conventionalized in their lexical meaning.

  The second two sentences have the same truth-conditions, yet also differ in one word: *couch* versus *sofa*. Therefore, *couch* and *sofa* must have the same truth-conditional meaning, and this is also conventionalized in their lexical meaning.
- **Pragmatic enrichment**

  Pragmatic enrichment is a phenomenon in which non-conventionalized aspects of meaning seem to contribute to truth-conditions of utterances.

  (3) I have not eaten. [*today*]

  In terms of its conventional meaning, this sentence just seems to express the proposition that it is not the case that the speaker has eaten anything before the utterance time. In most contexts, this would obviously be false, but the sentence is taken instead to be true, based on the pragmatically enriched proposition *I have not eaten today*. Pragmatic enrichment thus seems to have truth-conditional effects.

  Nevertheless, it is not conventional, as it is possible to come up with fantastical contexts in which the sentence does not have the enriched meaning. For example, consider a scenario in which babies are born linguistically mature, with the ability to utter sentences like the above. Suppose the sentence is uttered by a new-born baby. Then the enriched proposition would instead mean something like *I have not eaten ever* (assuming that part of the meaning of *eat* is to ingest through the mouth).

This example also illustrates the non-conventionality of pragmatic enrichment in another way. It is part of the conventional meaning of *eat* that when its object is dropped it must denote food (whatever counts as food for the agent). For example suppose the speaker is a secret agent who earlier today had to dispose of a top secret note by eating it, but who has not eaten anything else today. The speaker could then truthfully utter *I have not eaten*. Thus *food* is part of the conventional meaning of *eat* when its object is omitted, but not *today*. The latter is therefore a case of pragmatic enrichment, while the former is part of the conventional meaning or lexical semantics of *eat*. The sentence above therefore actually conventionally expresses the proposition *I have not eaten (food)* and is enriched suitably to *I have not eaten (food) today* (or in the weird baby context, *I have not eaten (food) ever*).

- **Conversational implicatures**

  Conversational implicatures are calculated based on context and real-world knowledge and are neither conventional nor truth-conditional.

  (4) Kim: What happened to my peanut butter sandwich?
  Sandy: I just saw the dog furiously licking its lips.
  $+>$ The dog ate the sandwich.

  It is not part of the conventional, truth-conditional meaning of Sandy's utterance that *the dog ate the sandwich*, but nevertheless this meaning seems to be expressed, because of the cooperative principle, the conversational maxims, and knowledge about the habits of dogs [Grice, 1989]. But this information is dependent on the context, including Kim's question. If Kim had instead asked *What happened to my shoe?*, the take-away from the same response by Sandy would not be that the dog ate the sandwich (what sandwich?).

- **Use-conditional meaning**

  Use-conditional meaning is conventional, but not truth-conditional. Consider these examples from Gutzmann [2015, 4] (the first pair following Frege).

  (5) This dog howled the whole night.

  (6) This cur howled the whole night.

  (7) This damn dog howled the whole night.

  Cur is in some sense a synonym for *dog*: the first two sentences do not have different truth conditions. However, there is an extra element of negative speaker attitude in the second, shared with the third, that is absent in the first.[1]

  A key piece of evidence that the negativity expressed by *cur* and *damn dog* are conventional is that every use of these expressions expresses this negativity on the part of the speaker. For example, if it is known that the speaker has nothing but positive feelings towards dogs, an utterance of a sentence like *Kim's cur/damn dog is here* would be quite odd.

  A key piece of evidence that the negativity is not truth-conditional is that it cannot be targeted by negation:

  (8) It's not true that this cur howled the whole night.

  (9) It's not true that this damn dog howled the whole night.

[1] It's a complicated question whether *cur* and *damn dog* are truly synonymous.

The negation here targets only the proposition that the dog howled the whole night and cannot cancel the negativity expressed. These sentences would simply be false if the dog indeed did howl the whole night but the speaker feels no antipathy towards it (they would also be odd, for the reason discussed in the previous paragraph).

2 Enriched Meanings

- What do we mean by enriched meanings?

  Broad definition: Any aspect of meaning that is not purely conventional and truth-conditional (descriptive meaning), but which can be computed in some manner from descriptive meaning. The broad definition therefore includes conventional implicature, conversational implicature, explicature (pragmatic enrichment), expressives, and presupposition.

  Narrow definition: A semantic representation that is derived from a more basic semantic representation and which potentially includes information that is not included in the basic representation. The space of enriched representations is nevertheless computed from the space of basic representations.

- The formal mechanism that we propose for capturing enriched meanings — at least narrowly construed — is *monads* [Moggi, 1989, 1990, Wadler, 1992, Wadler, 1994, 1995, Shan, 2001, Giorgolo and Unger, 2009, Giorgolo and Asudeh, 2011, 2012a,b, 2014a,b, Asudeh and Giorgolo, 2015, Charlow, 2014].[2]

- What are some intuitive advantages of our approach (compared to standard/popular approaches in linguistic theory)?

  1. A relatively simple compositional system: we need only add a single operator to a minimal compositional system.
  2. A conservative theory of lexical types: there is no generalization to the worst case, as in standard approaches, because only lexical items that encode instances of enriched meanings have enriched types.
  3. A variety of otherwise heterogeneous phenomena captured using a single set of formal tools: connections are made across phenomena that may otherwise not be apparent.
  4. An interdisciplinary exploration of the intersection of logic, language and computation: monads feature prominently in category theory and its applications, including functional programming.

[2]The references here are a mixture of our own work and the touchstone references that got us going on this path; this is not meant to be a list of the key works on monads in category theory or any such thing.

3 The Phenomena[3]

3.1 Multidimensionality: Expressives/conventional implicatures/use-conditional meaning

- Representing use-conditional meaning in a separate dimension from descriptive meaning explains why operators in the descriptive dimension cannot target use-conditional meaning.

(10) A: Most fucking neighbourhood dogs crap on my damn lawn.
B: No, that's not true.
$\Rightarrow$ No, the neighbourhood dogs don't crap on your lawn.
$\not\Rightarrow$ No, there's nothing wrong with dogs and/or their crapping on your lawn.

(11) A: John Lee Hooker, the bluesman from Tennessee, appeared in *The Blues Brothers*.
B: No, that's not true.
$\Rightarrow$ No, John Lee Hooker did not appear in *The Blues Brothers*.
$\not\Rightarrow$ No, John Lee Hooker was not from Tennessee.
B′: True, but actually John Lee Hooker was born in Mississipi

- In the foundational modern work in linguistic semantics on use-conditional meaning [Potts, 2005], information is standardly assumed to flow from the descriptive dimension (the *at-issue dimension* in the terminology of Potts 2005) to the use-conditional dimension (the *CI [conventional implicature] dimension* in the terminology of Potts 2005), but not vice versa. However, there seem to be exceptions [Potts, 2005, AnderBois et al., 2015, Giorgolo and Asudeh, 2011, 2012b].

(12) Mary, a good drummer, is a good singer too.

(13) Jake$_1$, who almost killed a woman$_2$ with his$_1$ car, visited her$_2$ in the hospital.

(14) Lucy, who doesn't help her sister, told Jane to.

(15) Melinda, who won three games of tennis, lost because Betty won six.

- It was initially argued that a lexical item contributes conventionally to either the descriptive dimension or the use-conditional dimension but not both [Potts, 2005], but there are lexical items that arguably contribute to both dimensions simultaneously [McCready, 2010, Gutzmann, 2015], such as slurs:

(16) They are Krauts.

(17) Your cur bit me.

3.2 Perspectives: Reference and substitution

- Co-referential terms sometimes cannot be substituted for each other (*Frege's Puzzle*):

(18) Hesperus is Phosphorus. *True*

(19) Kim believes that Hesperus is a planet. $\not\equiv$
(20) Kim believes that Phosphorus is a planet.

(21) Kim doesn't believe that Hesperus is Phosphorus. *Non-contradictory*

- Embedding under a propositional attitude verb is in fact not necessary for substitution puzzles to arise [Saul, 1997, 2007, Asudeh and Giorgolo, 2015].

(22) Clark Kent went into the phone booth, and Superman came out.

[3]The literature on each of these phenomena is large. For the purposes of these notes, I have not cited as extensively as I could have. Please consult the works cited for further references.

(23) Clark Kent went into the phone booth, and Clark Kent came out.

(24) # Dr. Octopus killed Spider-Man but he didn't kill Peter Parker.

(25) Dr. Octopus murdered Spider-Man but he didn't murder Peter Parker.

(26) Mary Jane loves Peter Parker, but she doesn't love Spider-Man.

- In such so-called 'simple sentences' [Saul, 1997, 2007], use of the same term in both relevant positions nevertheless is contradictory:

(27) # Dr. Octopus punched Spider-Man but he didn't punch Spider-Man.

- However, distinct terms are not necessary for substitution puzzles to arise [Giorgolo and Asudeh, 2014a, Asudeh and Giorgolo, 2015].

Context: Kim suffers from Capgras Syndrome, also known as the Capgras Delusion, a condition 'in which a person holds a delusion that a friend, spouse, parent, or other close family member has been replaced by an identical-looking impostor.'[4]

(28) Kim doesn't believe Sandy is Sandy.

Context: In 1897 Dr. Edwin J. Goodwin presented a bill to the Indiana General Assembly for '[...] introducing a new mathematical truth and offered as a contribution to education to be used only by the State of Indiana free of cost'. He had copyrighted that $\pi = 3.2$ and offered this 'new mathematical truth' for free use to the State of Indiana (but others would have to pay to use it).[5]

(29) Dr. Goodwin doesn't believe that π is π.

- We think that these cases can be unified in a general semantics of *perspectives*, using monads to capture the intuition formally [Giorgolo and Asudeh, 2014a, Asudeh and Giorgolo, 2015].

	Simple	**Embedded**
Same term	# Dr. Octopus punched Spider-Man but he didn't punch Spider-Man.	Kim doesn't believe Sandy is Sandy.
Distinct term	Mary Jane loves Peter Parker but she doesn't love Spider-Man. # Dr. Octopus killed Spider-Man but he didn't kill Peter Parker.	Kim doesn't believe Hesperus is Phosphorus.

Table 2: Overview of substitution puzzles.

3.3 Uncertainty: Reasoning fallacies

- It has famously been shown that subjects in psychological experiments sometimes fail to reason logically about probabilities [Tversky and Kahneman, 1983]. For example, under appropriate conditions, the majority of subjects in T&K's studies rated the likelihood of the conjunction of two events as higher than the likelihood of one of the conjoined events.

- Probability theory tells us:

[4]Wikipedia: `http://en.wikipedia.org/wiki/Capgras_delusion`
[5]Wikipedia: `http://en.wikipedia.org/wiki/Indiana_Pi_Bill`

(30) $P(A \,\&\, B) \leq P(A), P(B)$

- One example of this is the well-known "Linda paradox" [Tversky and Kahneman, 1983]. Subjects were given the following statement and, as part of the experimental task, where asked to rank the probability that various statements were true of Linda; the resulting ranking for the relevant cases is given below the context.

(31) Linda is 31 years old, single, outspoken and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.

Linda is active in the feminist movement. [F(eminist)]
Linda is a bank teller and is active in the feminist movement. [T&F]
Linda is a bank teller. [T(eller)]

- T&K's explanation of this, based on a heuristic of *representativeness*, is non-compositional [Giorgolo and Asudeh, 2014b]. Moreover, these sorts of results have been shown to be context-sensitive in systematic ways [Hertwig and Gigerenzer, 1999].
- These fallacies can receive an alternative explanation — one that is concordant with standard pragmatic theory [Grice, 1989, Levinson, 1983] — using two related *semirings* and monadic machinery to compose and store probabilistic measures associated with semantic types.
- I will not talk about this third case today, but for details see Giorgolo and Asudeh [2014b].

4 Case 1. Conventional Implicature/Multidimensional Semantics

- The main ingredients of our analysis of conventional implicature are the notion of *paired semantic values* as introduced by Potts [2005] and the monadic interface.
- According to Potts, expressions conveying conventional implicatures denote two semantic objects: an at-issue value (which is often empty and corresponds to the identity function) and a side-issue component which is always a proposition.
- This assumption will also form the core of our approach. The difference is that in our approach all expressions are interpreted as denoting a pair of values.
  - The first component of the pair denotes the at-issue contribution of the expression.
  - The second component is not a proposition but rather a *collection of propositions*, containing all the side-issue information conveyed by the sub-parts from which the expression is composed.
  - Expressions without conventional implicature-bearing items denote an empty collection of CI propositions.
- The monadic framework allows us to reuse the standard compositional machinery to compose these more complex meanings, while controlling the flow of information as desired.
- The monad we use is constructed around the functor that maps each type τ to the type of pairs of elements of τ and elements of a monoid M.
  - When we want to be precise we write down this functor as $\Diamond_{ci}$, but given that we are working with a single functor we will drop the subscript and just write $\Diamond$.
  - The monoid we will use is the powerset of P, the set of all propositions with set union as the binary operation and $\emptyset$ as the identity element.
  - The second component of the functor maps every function f to a lifted version that operates on the first component of our pairs.

- Abusing the syntax of the lambda calculus a little bit we can write it down as:

$$\Diamond(f) = \lambda\langle x, Q\rangle.\langle f(x), Q\rangle \tag{32}$$

- The monoid functions as a storage for side-issue propositions, accumulating them in a set.
- The identity $\emptyset$ corresponds to the case when no side-issue comment has been made, while union is used to accumulate semantic material.

• The second ingredient for our monad is the "unit" natural transformation.

- The unit will map each object (i.e. type) in our category to an arrow in the same category that goes from the same object to the result of applying the functor to that object.
- In our case we will have a family of functions that map each element of a type to the pair composed of the same element together with the monoid identity, or formally:

$$\eta_A(x) = \langle x, \emptyset\rangle \tag{33}$$

• Let's check that this is indeed a natural transformation. To do so we need to check that the following diagram commutes:

$$\begin{tikzcd}
Id(A) \arrow[r, "\eta_A"] \arrow[d, "Id(f)"'] & \Diamond A \arrow[d, "\Diamond(f)"] \\
Id(B) \arrow[r, "\eta_B"'] & \Diamond B
\end{tikzcd}$$

and here is the proof that this is the case:

$$\eta_B(f(x)) = \langle f(x), \emptyset\rangle = \Diamond(f)(\langle x, \emptyset\rangle) = \Diamond(f)(\eta_A(x))$$

- Intuitively, the unit takes a value and turns it into something with the type of an expression contributing a conventional implicature, although the contribution is in this case empty (which is exactly what we want).

• Now we can either introduce join or bind. Let's do both.

- Remember that join is a natural transformation that maps to each object a function that goes from double application of the functor to a single one.
- In other words for a type τ it should give us a function of type $\Diamond\Diamond\tau \to \Diamond\tau$, or by expanding the definition of our functor $\tau \times \mathcal{P}(P)) \times \mathcal{P}(P) \to \tau \times \mathcal{P}(P)$.
- Given that the second component of our pairs is an element of a monoid we can combine two of them by using the binary operation of the monoid, so our join transformation μ will assign to each object A the following function:

$$\mu_A(\langle\langle x, P\rangle, Q\rangle) = \langle x, P \cup Q\rangle \tag{34}$$

- Let's check again that this is indeed a natural transformation:

$$\begin{tikzcd}
\Diamond\Diamond A \arrow[r, "\mu_A"] \arrow[d, "\Diamond\Diamond(f)"'] & \Diamond A \arrow[d, "\Diamond(f)"] \\
\Diamond\Diamond B \arrow[r, "\mu_B"'] & \Diamond B
\end{tikzcd}$$

and here is the proof that the diagram commutes:

$$\begin{aligned}
&\mu_B(\Diamond\Diamond(f)(\langle\langle x, P\rangle, Q\rangle)) = \\
&\mu_B(\langle\langle f(x), P\rangle, Q\rangle) = \\
&\langle f(x), P \cup Q\rangle = \\
&\Diamond(f)(\langle x, P \cup Q\rangle) = \\
&\Diamond(f)(\mu_A(\langle\langle x, P\rangle, Q\rangle))
\end{aligned}$$

- To show that this is indeed a monad we need also to prove that following two diagrams commute for all objects:

$$\begin{tikzcd}
\Diamond A \arrow[r, "\eta_{\Diamond A}"] \arrow[dr, "id"'] & \Diamond\Diamond A \arrow[d, "\mu_A"] & \Diamond A \arrow[l, "\Diamond(\eta_A)"'] \arrow[dl, "id"] \\
& \Diamond A &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Diamond\Diamond\Diamond A \arrow[r, "\mu_{\Diamond A}"] \arrow[d, "\Diamond(\mu_A)"'] & \Diamond\Diamond A \arrow[d, "\mu_A"] \\
\Diamond\Diamond A \arrow[r, "\mu_A"'] & \Diamond A
\end{tikzcd} \tag{35}$$

  - We leave the proof as an exercise. Instead we derive the definition of bind from μ and show that the three monad axioms hold.
  - In general bind can be defined in terms of join as follows:

$$m \star f = \mu(\Diamond(f)(m)) \tag{36}$$

  - In our case we obtain the following definition:

$$\langle x, Q\rangle \star f \rightsquigarrow \mu(\langle f(x), Q\rangle) \rightsquigarrow \mu(\langle\langle y, P\rangle, Q\rangle \rightsquigarrow \langle y, P \cup Q\rangle \tag{37}$$

    where $\langle y, P\rangle$ is the result of applying f to x (remember that the type of f is in general $A \to \Diamond B$).
  - Another way to write this definition using projections is the following (in this case we don't have to explicitly give names to the result of $f(x)$:

$$\langle x, P\rangle \star f = \langle \pi_1(f(x)), P \cup \pi_2(f(x))\rangle \tag{38}$$

- To show that this is indeed a monad we prove that our definitions respect the three monad rules:

$$m \star \eta = m \tag{39}$$
$$\eta(x) \star f = f(x) \tag{40}$$
$$(m \star f) \star g = m \star (\lambda x. f(x) \star g) \tag{41}$$

  1. The first is easy:

$$\langle x, P\rangle \star \eta = \langle \pi_1(\eta(x)), P \cup \pi_2(\eta(x))\rangle = \langle \pi_1(\langle x, \emptyset\rangle), P \cup \pi_2(\langle x, \emptyset\rangle)\rangle = \langle x, P \cup \emptyset\rangle = \langle x, P\rangle$$

  2. Here is the proof for the second rule:

$$\eta(x) \star f = \langle x, \emptyset\rangle \star f = \langle \pi_1(f(x)), \emptyset \cup \pi_2(f(x))\rangle = \langle \pi_1(f(x)), \pi_2(f(x))\rangle = f(x)$$

  3. For the proof of the final rule let's use this equivalent definition for bind:

$$m \star f = \langle \pi_1(f(\pi_1(m))), \pi_2(m) \cup \pi_2(f(\pi_1(m)))\rangle \tag{42}$$

     the only difference being that here we decompose the first argument using projection functions instead of our sloppy pattern matching.

(a) First let's see what the left hand side of the rule reduces to:

$$
\begin{aligned}
&(m \star f) \star g = \\
&\langle \pi_1(f(\pi_1(m))), \pi_2(m) \cup \pi_2(f(\pi_1(m))) \rangle = \\
&\langle \pi_1(g(\pi_1(f(\pi_1(m))))), \pi_2(m) \cup \pi_2(f(\pi_1(m))) \cup \pi_2(g(\pi_1(f(\pi_1(m))))) \rangle
\end{aligned}
$$

(b) And here is the right hand side reduction:

$$
\begin{aligned}
&m \star (\lambda x.f(x) \star g) = \\
&\langle \pi_1(f(\pi_1(m)) \star g), \pi_2(m) \cup \pi_2(f(\pi_1(m)) \star g) \rangle = \\
&\langle \pi_1(\langle \pi_1(g(\pi_1(f(\pi_1(m))))), \pi_2(f(\pi_1(m))) \cup \pi_2(g(\pi_1(f(\pi_1(m))))) \rangle), \\
&\pi_2(m) \cup \pi_2(\langle \pi_1(g(\pi_1(f(\pi_1(m))))), \pi_2(f(\pi_1(m))) \cup \pi_2(g(\pi_1(f(\pi_1(m))))) \rangle) \rangle = \\
&\langle \pi_1(g(\pi_1(f(\pi_1(m))))), \pi_2(m) \cup \pi_2(f(\pi_1(m))) \cup \pi_2(g(\pi_1(f(\pi_1(m))))) \rangle
\end{aligned}
$$

The two reductions end in the same term as expected.

4.1 Analysis

- Let's apply our machinery to one of our early examples:

  (43) John Lee Hooker, the bluesman from Tennessee, appeared in The Blues Brothers

- The reading we expect is one in which the proposition that JLH appeared in TBB movie is the at-issue contribution, while the fact that he is from Tennessee is part of the side-issue comments.
- In our formalism the first proposition will be the first component of the meaning pair, while the side-issue comment will be the only element of the set of propositions in the second projection of the pair (it's the only element because there are no other conventional implicatures involved in this example).
- The mini lexicon we use is show in table 3.
- We assume fairly standard lexical entries, the only entries that probably need some comment are those for *the* and *COMMA*.
- The entry for *the* is not probably what you would expect, but arguably *the* here doesn't so much play the role of a definite article, as it could be replaced by the expression *who is a* without changing much the final meaning of the entire sentence.
- We choose to treat its meaning simply as an identity function that at the level of syntax just changes the type to the preferred one (the one of a non-restrictive relative clause).[6]
- In this way we can provide a single lexical entry for the prosodic element *COMMA* that works both for cases involving a definite article and full non-restrictive relative clauses.
- In any case, notice that this choice is necessary not so much because of the way we set up our system for dealing with conventional implicatures but more because of the way the grammar of the language has to be specified.
- The interesting part of the lexicon is the denotation of *COMMA*.
  - Its denotation is a function that takes as arguments the property expressed by the parenthetical and the referent associated with the NP the comment is about.
  - The function then generates a monadic value.

[6] Bare nouns seems to be less common as parentheticals but they are not completely out: *Guybrush Threepwood, mighty pirate, is the proud owner of a rubber chicken with a pulley in the middle.*

- First of all the application of the "parenthetical" predicate to the referent is added to the side-issue dimension using the function `write` which can be defined as follows:

$$\mathtt{p} = \langle 1, \{p\} \rangle \tag{44}$$

- The function takes a proposition and returns a pair formed by the dummy value of our terminal type 1 and the singleton set containing the passed proposition.
- Finally the denotation of *COMMA* returns the referent passed as an argument as its core value, lifting it to a monadic value using η.
- Notice that bind takes care of carrying over the side-issue proposition introduced by `write`, while the dummy value 1 is discarded by the vacuous abstraction after $\star$.

WORD	SYNTACTIC TYPE	DENOTATION	SEMANTIC TYPE
John Lee Hooker	np	jlh	e
Tennessee	np	tn	e
The Blues Brothers	np	tbb	e
bluesman	n	$\lambda x.bluesman(x)$	$e \to t$
the	$(np\backslash s)/n$	$\lambda x.x$	$(e \to t) \to e \to t$
from	$(n\backslash n)/np$	$\lambda x.\lambda P.\lambda y.P(y) \wedge from(y, x)$	$e \to (e \to t) \to e \to t$
appeared in	$(np\backslash s)/np$	$\lambda x.\lambda y.appearedIn(y, x)$	$e \to e \to t$
COMMA	$(np\backslash\Diamond np)/(np\backslash s)$	$\lambda P.\lambda x.\mathtt{write}(P(x)) \star \lambda y.\eta(x)$	$(e \to t) \to e \to \Diamond e$

Table 3: Toy lexicon.

- Now we are ready to see how our system generates the expected reading for our example.
- The derivation itself does not fit easily in the format of a page, so we instead zoom in on how the generated reading gets reduced to the expected pair.
- The reading we get is the following:

  $$[\![\text{COMMA}]\!]\,([\![\text{the}]\!]\,([\![\text{from}]\!]\,([\![\text{Tennessee}]\!])([\![\text{bluesman}]\!])))([\![\text{JLH}]\!]) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w))$$

  Let's what happens when we substitute our lexical entries:

  $$\begin{aligned}
  &[\![\text{COMMA}]\!]\,([\![\text{the}]\!]\,([\![\text{from}]\!]\,([\![\text{Tennessee}]\!])([\![\text{bluesman}]\!])))([\![\text{JLH}]\!]) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &[\![\text{COMMA}]\!]\,([\![\text{the}]\!]\,(\lambda y.bluesman(y) \wedge from(y, tn)))([\![\text{JLH}]\!]) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &[\![\text{COMMA}]\!]\,(\lambda y.bluesman(y) \wedge from(y, tn))([\![\text{JLH}]\!]) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &(\mathtt{write}(bluesman(jlh) \wedge from(jlh, tn)) \star \lambda y.\eta(jlh)) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &(\langle 1, \{bluesman(jlh) \wedge from(jlh, tn)\}\rangle \star \lambda y.\langle jlh, \emptyset\rangle) \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &\langle jlh, \{bluesman(jlh) \wedge from(jlh, tn)\}\rangle \star \lambda w.\eta([\![\text{appeared in}]\!]\,([\![\text{TBB}]\!])(w)) \rightsquigarrow\\
  &\langle jlh, \{bluesman(jlh) \wedge from(jlh, tn)\}\rangle \star \lambda w.\langle appearedIn(w, tbb), \emptyset\rangle \rightsquigarrow\\
  &\langle appearedIn(jlh, tbb), \{bluesman(jlh) \wedge from(jlh, tn)\}\rangle
  \end{aligned}$$

	Simple	Embedded
Same term	# Dr. Octopus punched Spider-Man but he didn't punch Spider-Man.	Kim doesn't believe Sandy is Sandy.
Distinct term	Mary Jane loves Peter Parker but she doesn't love Spider-Man. # Dr. Octopus killed Spider-Man but he didn't kill Peter Parker.	Kim doesn't believe Hesperus is Phosphorus.

Table 4: The space of explananda.

5 Case 2. Substitution Puzzles: Perspectives

- Before showing our proposal, it might be useful to the group to see an alternative formalization done in the style of the Logical Form semantics of Heim and Kratzer [1998].
- This shows what a linguist might more standardly be tempted to do (it's yucky) and will provide a baseline against which we argue that our solution is preferable.

5.1 A Non-Monadic Formalization

- Our analysis depends crucially on the availability of different points of view during the interpretation process.
- One simple formalization of this idea is to make the interpretation function that maps expressions to meanings have an additional parameter representing a perspective.
- Therefore, in order to interpret an expression α, we will need both an assignment function (as is standard) and a *perspective index*.
- We represent the interpretation of an expression α as $[\![\alpha]\!]^{g,i}$, where g is the assignment function and i the perspective index.
- To get a compositional system we also need a way to represent *application* and *abstraction*.
- In both cases we simply want the perspective indices to be left untouched by the compositional process, as according to our analysis all changes in perspective are determined by the lexicon.
- The revised form of application [Heim and Kratzer, 1998, 105] is defined in (45); the perspective index for the interpretation of the composed expression is the same as that of its subexpressions.

(45) *Revised Application Rule:* Let α be a branching node with daughters β and γ. Then for any assignment function g and perspective index i, $[\![\alpha]\!]^{g,i} = [\![\beta]\!]^{g,i}\,([\![\gamma]\!]^{g,i})$ or $[\![\alpha]\!]^{g,i} = [\![\gamma]\!]^{g,i}\,([\![\beta]\!]^{g,i})$, as determined by the semantic types of β and γ.

- Similarly, in the case of the Predicate Abstraction rule, the interpretation index is carried over in the body of the lambda abstraction.
- In (46) we present a revised version of the rule as discussed in Heim and Kratzer [1998, 186].

(46) *Revised Predicate Abstraction Rule:* Let α be a branching node with daughters β and γ, where β dominates only a numerical index j. Then, for any variable assignment g and perspective index i, $[\![\alpha]\!]^{g,i} = \lambda x.\, [\![\gamma]\!]^{g^{x/j},i}$, where $g^{x/j}$ is the same assignment function as g except that it maps x to j.

- In such a system all expressions are interpreted with respect to a perspective index.
- In most cases such indices are not used for determining the denotation of an expression.
- For instance, for a name that the speaker understands to be non-controversial, such as *Mary Jane* or *Peter Parker*, the interpretation is fixed and independent of a perspective:

$$[\![\text{Mary Jane}]\!]^{g,i} = \mathbf{mj}_\sigma \tag{47}$$

$$[\![\text{Peter Parker}]\!]^{g,i} = \mathbf{pp}_\sigma \tag{48}$$

- On the other hand, in the case of a name whose interpretation is contentious between different speakers, the final denotation is based on the perspective index passed to the interpretation step.
- So the name *Spider-Man* will have a denotation that depends on the perspective taken during the interpretation process.
- For our speaker well-versed in the Spider-Man universe, the name will denote the same entity as *Peter Parker*, while for Mary Jane the same name will denote a different entity:[7]

$$[\![\text{Spider-Man}]\!]^{g,i} = \begin{cases} \mathbf{sm}_{\texttt{mj}} \text{ if } i = \texttt{mj} \\ \mathbf{pp}_\sigma \text{ if } i = \sigma \end{cases} \tag{49}$$

- The denotation for the verb *love* is slightly different, as it involves a direct manipulation of the perspective indices which are part of the interpretational meta-language.[8]
- In the case of *love* we want to be able to force the perspective index of the expression in the object position to be the perspective of (the denotation of) the subject of the verb.
- Given that we can manipulate the perspective indices only at the level of the interpretational meta-language, the denotation for *love* needs to include as an argument the expression in the object position, rather than the denotation of the object itself (κ is a function that maps entities to perspective indices; see below):

$$[\![\text{loves DP}]\!]^{g,i} = \lambda s.\mathbf{love}(s, [\![\text{DP}]\!]^{g,\kappa(s)}) \tag{50}$$

- In contrast, in the case of a different transitive verb, like *punch*, which does not involve a potential switch in perspective, we provide a denotation that operates entirely at the level of the meaning language:

$$[\![\text{punch}]\!]^{g,i} = \lambda o.\lambda s.\mathbf{punch}(s, o) \tag{51}$$

- Equipped with this mini lexicon, we can sketch a preliminary analysis of an example like (**??**), repeated here as (52)

  (52) Mary Jane loves Peter Parker, but she doesn't love Spider-Man.

  Our analysis is centered around the fact that (52) has a non-contradictory reading because the object of the second conjunct is not necessarily assigned the same denotation as the object of the first conjunct.

[7]Note that it is not important for our account that the speaker's denotation for both *Peter Parker* and *Spider-Man* is "$\mathbf{pp}_\sigma$" as such, but rather just that 1. Mary Jane and the speaker's denotations are the same for *Peter Parker*; 2. are not the same for *Spider-Man*; and 3. the speaker's denotations are the same for *Peter Parker* and *Spider-Man*.

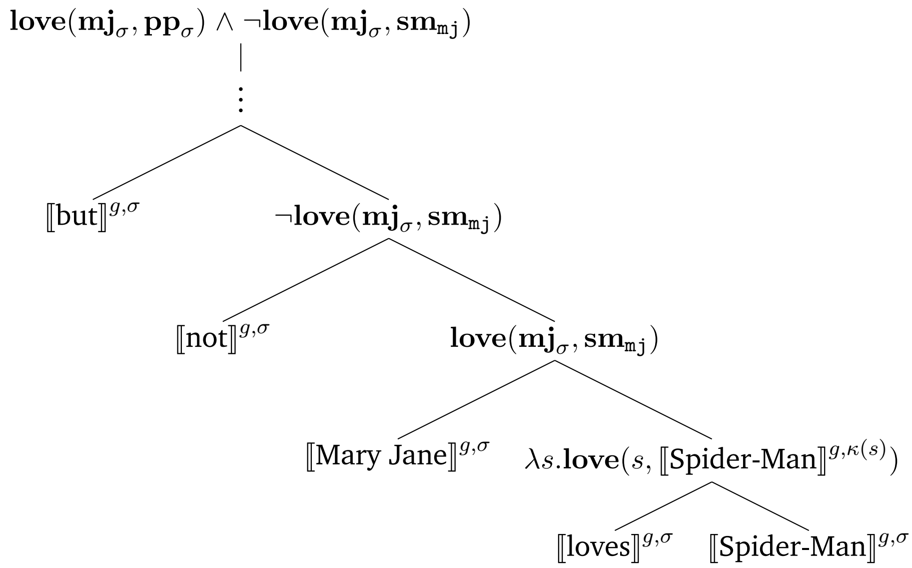
[8]In principle we could add the indices to the target meaning language, and this is indeed the choice we make in our alternative monadic implementation below. However in the case of standard Heim and Kratzer-style semantics we would still need to modify the rules for functional application and predicate abstraction, as otherwise the types of the denotations would not match properly.

- We expect to have two readings, one contradictory and one instead consistent with what the enlightened know about the Spider-Man universe.

- The two readings correspond to two different scopal relationships between the proper names.

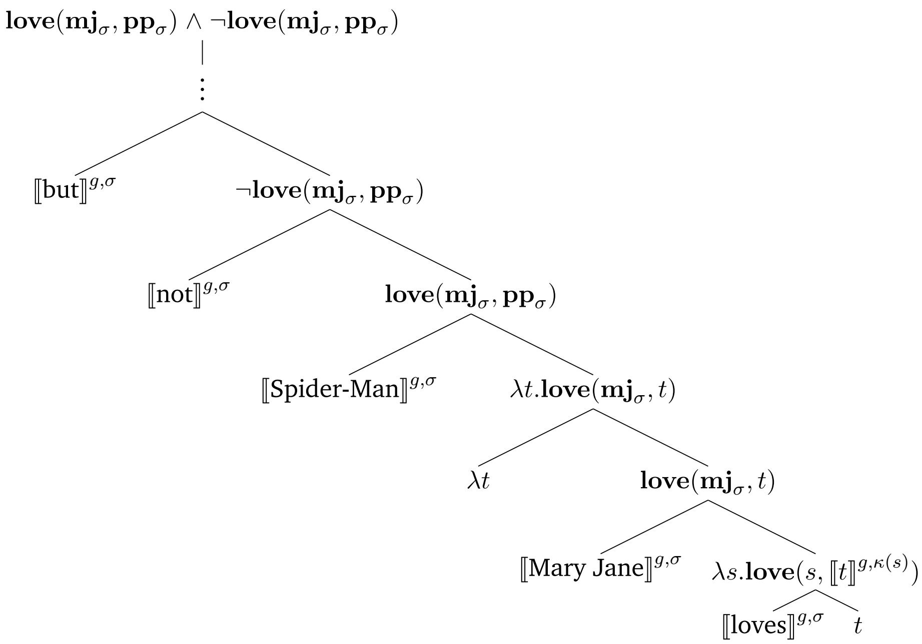
- In the case of the consistent reading, the name *Spider-Man* is evaluated in the scope of the verb *love* and therefore is interpreted from the perspective of Mary Jane:

(53)



- In the case of the contradictory reading, the name *Spider-Man* is instead interpreted from the perspective of the speaker, who, according to our assumptions, knows his secret identity:

(54)



- Notice that, for this last interpretation to work, we have to stipulate that traces are not interpretable, or rather that they evaluate to themselves in all cases.

- There are a number of reasons why we think that the monadic approach we will introduce below is preferable to the Logical Form semantics that we have just sketched.

  1. In our monadic account we are not forced to generalize the lexicon to the worst case, introducing perspective indices everywhere. Indices are instead introduced in the derivation only if needed and the process is entirely governed by the compositional logic, instead of being a generalized lifting of the lexicon.

2. In turn this means that we do not need to modify the rule for functional application: since indices are introduced in the derivation, their propagation is controlled by the specific part of the logic that deals with monads, together with special lexical specifications, such as the one for verbs like *love* or *believe*.
3. At the same time we do not need to introduce syncategorematic rules for interpreting special expressions as we were forced to do in the present setting for the verb *love*. The distinction between the interpretational meta-language and the target language is still present in the monadic approach but we have a much more constrained way of bridging the two levels thanks to the monadic operations.
4. Another reason why we believe that the monadic approach is preferable is its generality. We can in fact reuse the same compositional mechanism to account for a variety of semantic phenomena, as pointed out by Shan [2001] and as further investigated in various other works [Giorgolo and Unger, 2009, Unger, 2011, Giorgolo and Asudeh, 2011, 2012b,a, 2014a,b, Charlow, 2014]. In other words, the monadic approach makes more evident a general pattern of enhanced composition that is otherwise hard-wired in the system by generalized type lifts and alternative compositional rules.

5.2 Formalization with Monads

- We will use the monad that describes values that are made dependent on some external parameter, commonly known in the functional programming literature as the Reader monad.
- This follows Shan [2001], who suggested the idea of using the Reader monad to model intensional phenomena in natural language.
- We will represent linguistic expressions that can be assigned potentially different interpretations as functions from perspective indices to values.
- Effectively we will construct a kind of lexicon that not only represents the linguistic knowledge of a single speaker but also her (possibly partial) knowledge of the language of other speakers.
- In other words, we construe lexicons to be aspects of the knowledge of language of *individuals*, and take standard circumlocutions like the "lexicon of English" to be atheoretical folk talk, if not simply incoherent. This is a well-established position in generative linguistics [Chomsky, 1965, 1986, 2000, Jackendoff, 1983, 1997, 2002, 2007].
- So we claim that examples like the Capgras example or the similar following example can be assigned non-contradictory readings:[9]

  (55) Reza doesn't believe Jesus is Jesus.

- The speaker's lexicon also includes the information regarding Reza's interpretation of the name *Jesus* and therefore makes it possible for the speaker to use the same expression, in combination with a verb such as *believe*, to actually refer to two different entities.
- In one case we will argue that the name *Jesus* is interpreted using the speaker's perspective while in the other case it is Reza's perspective that is used.

[9]This example is based on the controversy from the summer of 2013 in which the scholar Reza Aslan was taken to task by Fox News correspondent Lauren Green for his views about the historical figure of Jesus of Nazareth. It seems to us that (55) could have been said sincerely by Green in that context. `http://en.wikipedia.org/wiki/Reza_Aslan`

5.2.1 A Monad for Perspectives

- To avoid introducing the complexities of the categorical formalism, we introduce monads as they are usually encountered in the computer science literature, as in our previous work [Giorgolo and Asudeh, 2014a].
- A monad is defined as a triple $\langle \Diamond, \eta, \star \rangle$. $\Diamond$ is a *functor*, in our case a mapping between types and functions.
- We call the component of $\Diamond$ that maps between types $\Diamond_1$ and the one that maps between functions $\Diamond_2$.
- In our case, $\Diamond_1$ will map each type to a new type that corresponds to the original type with an added perspective index parameter.
- Formally, if i is the type of perspective indices, then $\Diamond_1$ maps any type τ to $i \rightarrow \tau$. The functor $\Diamond_2$ maps any function $f : \tau \rightarrow \delta$ to a function $f' : (i \rightarrow \tau) \rightarrow i \rightarrow \delta$. $\Diamond_2$ corresponds to function composition:
$$\Diamond_2(f) = \lambda g.\lambda i.f(g(i)) \tag{56}$$
- The component $\Diamond_2$ will not be used below, so we will use $\Diamond$ as an abbreviation for $\Diamond_1$. This means that we will write $\Diamond\tau$ for the type $i \rightarrow \tau$.
- η ('unit') is a polymorphic function that maps inhabitants of a type τ to inhabitants of its image under $\Diamond$, formally $\eta : \forall\tau.\tau \rightarrow \Diamond\tau$.
- Using the computational metaphor, η should embed a value in a computation that returns that value without any side-effect.
- In our case η should simply add a vacuous parameter to the value:
$$\eta(x) = \lambda i.x \tag{57}$$
- $\star$ ('bind') is a polymorphic function of type $\forall\tau.\forall\delta.\Diamond\tau \rightarrow (\tau \rightarrow \Diamond\delta) \rightarrow \Diamond\delta$, and acts as a sort of enhanced functional application.[10]
- Again using the computational metaphor, $\star$ takes care of combining the side effects of the argument and the function and returns the resulting computation.
- In the case of the monad that we are interested in, $\star$ is defined as in (58).
$$a \star f = \lambda i.f(a(i))(i) \tag{58}$$
- Another fundamental property of $\star$ is that, by imposing an order of evaluation, it provides us with an additional scoping mechanism distinct from standard functional application.
- This will allow us to correctly capture the multiple readings associated with the expressions under consideration.
- Every monad defined in terms of unit and bind must satisfy the monad laws introduced above, in the discussion of conventional implicature:
$$\eta(x) \star f = f(x) \tag{59}$$
$$m \star \eta = m \tag{60}$$
$$(m \star f) \star g = m \star \lambda x.(f(x) \star g) \tag{61}$$

[10] We use the argument order for $\star$ that is normally used in functional programming, rather than swapping the arguments to make it look more like standard functional application, which would be an alternative, equivalent notational choice. We write $\star$ in infix notation.

- The first two axioms guarantee that unit behaves as a "multiplicative unit" with respect to bind, while the last axiom is a form of associativity, i.e. only the linear order of the monads combined with bind matters, not their grouping into a tree structure. Our monad satisfies the three axioms.

- In sum, we add two operators, η and $\star$, to the lambda calculus and the reductions work as expected for (57) and (58). These reductions are implicit in our analyses below.

5.3 Analysis

- We will exemplify our approach with analyses of a selection of the following examples:

(62) Kim doesn't believe Hesperus is Phosphorus.

(63) # Dr. Octopus punched Spider-Man but he didn't punch Spider-Man.

(64) Mary Jane loves Peter Parker but she doesn't love Spider-Man.

(65) Kim doesn't believe Sandy is Sandy.

- Example (64) is to be understood given the context that MJ does not know Peter Parker's secret and example (65) is to be understood in a Capgras context.

- The starting point for our analysis of these examples is the lexicon in Table 5.

- The lexicon represents the linguistic knowledge of the speaker, including her knowledge of other individuals' grammars.[11]

[11] We have simplified some entries in Table 5 by writing, e.g., '$\mathbf{ms}_{\mathtt{k}}$ if $i = \mathtt{k}$' instead of '$\mathbf{ms}_i$ if $i = \mathtt{k}$', where there are not multiple options for i. For example, contrast the entry of *Phosphorus* with that of *Spider-Man*.

WORD	DENOTATION	CATEGORY	TYPE
Reza	$\mathbf{r}_\sigma$	np	e
Kim	$\mathbf{k}_\sigma$	np	e
Dr. Octopus	$\mathbf{o}_\sigma$	np	e
Mary Jane	$\mathbf{mj}_\sigma$	np	e
Peter Parker	$\mathbf{pp}_\sigma$	np	e
not	$\lambda P.\lambda x.\neg P(x)$	$(np\backslash s)/(np\backslash s)$	$(e \to t) \to (e \to t)$
but	$\lambda p.\lambda q.p \wedge q$	$(s\backslash s)/s$	$t \to t \to t$
is	$\lambda x.\lambda y.x = y$	$(np\backslash s)/np$	$e \to e \to t$
punch	$\lambda o.\lambda s.\mathbf{punch}(s, o)$	$(np\backslash s)/np$	$e \to e \to t$
believe	$\lambda c.\lambda s.\mathbf{B}(s, c(\kappa(s)))$	$(np\backslash s)/\Diamond s$	$\Diamond t \to e \to t$
love	$\lambda o.\lambda s.\mathbf{love}(s, o(\kappa(s)))$	$(np\backslash s)/\Diamond np$	$\Diamond e \to e \to t$
Hesperus	$\lambda i.\begin{cases}\mathbf{es}_{\mathtt{k}} & \text{if } i = \mathtt{k},\\ \mathbf{v}_\sigma & \text{if } i = \sigma\end{cases}$	$\Diamond np$	$\Diamond e$
Phosphorus	$\lambda i.\begin{cases}\mathbf{ms}_{\mathtt{k}} & \text{if } i = \mathtt{k},\\ \mathbf{v}_\sigma & \text{if } i = \sigma\end{cases}$	$\Diamond np$	$\Diamond e$
Spider-Man	$\lambda i.\begin{cases}\mathbf{sm}_i & \text{if } i = \mathtt{o} \text{ or } i = \mathtt{mj},\\ \mathbf{pp}_\sigma & \text{if } i = \sigma\end{cases}$	$\Diamond np$	$\Diamond e$
Jesus	$\lambda i.\begin{cases}\mathbf{j}_{\mathtt{r}} & \text{if } i = \mathtt{r},\\ \mathbf{j}_\sigma & \text{if } i = \sigma\end{cases}$	$\Diamond np$	$\Diamond e$
Sandy	$\lambda i.\begin{cases}\mathbf{imp}_{\mathtt{k}} & \text{if } i = \mathtt{k},\\ \mathbf{s}_\sigma & \text{if } i = \sigma\end{cases}$	$\Diamond np$	$\Diamond e$

Table 5: Speaker's lexicon.

- Most lexical entries are standard, since we do not have to generalize to the worst case.
- So we do not need to change the type and denotation of lexical items that are not involved in the phenomena under discussion.
- For instance, logical operators such as *not* and *but* are interpreted in the standard way, as is a verb like *punch* or *kill*.
- Referring expressions that are possibly contentious, in the sense that they can be interpreted differently by the speaker and other individuals, instead have the monadic type $\Diamond e$.[12]
- This is reflected in their denotation by the fact that their value varies according to a perspective index.
- We use a special index σ for the speaker's own perspective, and assume that this is the default index used whenever no other index is specifically introduced.
- For example, in the case of the name *Spider-Man*, we are assuming that the speaker is aware of his secret identity and therefore interprets it as another name for the individual Peter Parker,[13] while Mary Jane and Dr. Octopus consider Spider-Man to be a different entity from Peter Parker.
- The other special lexical entries in our lexicon are those for verbs like *believe* and *love*.
- The two entries are similar in the sense that they both take an already monadic resource and actively supply a specific perspective index that corresponds to the subject of the verb.
- The function κ maps each entity to the corresponding perspective index, i.e.:

$$\kappa : e \to i \tag{66}$$

- κ is defined for the relevant cases under consideration as follows:

$$\kappa(\mathbf{r}_\sigma) = \mathtt{r} \tag{67}$$
$$\kappa(\mathbf{k}_\sigma) = \mathtt{k} \tag{68}$$
$$\kappa(\mathbf{o}_\sigma) = \mathtt{o} \tag{69}$$
$$\kappa(\mathbf{mj}_\sigma) = \mathtt{mj} \tag{70}$$

- In the lexical entries for *believe* and *love*, κ maps the subject to the perspective index of the subject.
- Thus, the entry for *believe* uses the subject's point of view as the perspective used to evaluate its entire complement, while *love* changes the interpretation of its object relative to the perspective of its subject.
- However we will see that the interaction of these lexical entries and the evaluation order imposed by $\star$ will allow us to let the complement of a verb like *believe* and the object of a verb like *love* escape the specific effect of forcing the subject perspective, and instead we will be able to derive readings in which the arguments of the verb are interpreted using the speaker's perspective.

[12]It may be that there is an equivalence between these sorts of contentious expressions in our system and the *restricted names* of Zimmermann [2005] and between our non-contentious expressions and his *neutral names*, but the formal details are sufficiently different that the equivalence is not immediately obvious. Moreover, Zimmermann's distinction is restricted to names, but we show in section **??** that our solution is more general than this.

[13]See footnote 7 for some further clarification of this point.

- Figure 1 shows the four non-equivalent readings that we derive in our system for example (62), repeated here as (72).[14]

  (72) Kim doesn't believe that Hesperus is Phosphorus.

- Notice that the type of the proof goal is $\Diamond t$ despite the fact that the result type of the predicate of the main clause, *believe*, is t.
- This is in general necessary if the sentence includes at least a single linguistic resource with a positive instance of a monadic type.
- We can always deal with this thanks to the $\Diamond R$ rule.[15]
- Reading (77) assigns to both *Hesperus* and *Phosphorus* the subject Kim's interpretation and results, after contextualising the sentence by applying it to the standard σ perspective index, in the truth conditions in (73), i.e. that Kim does not believe that Hesperus *qua* the evening star is Phosphorus *qua* the morning star.
- This reading would not be contradictory in an epistemic model (such as Kim's model) where the evening star and the morning star are not the same entity.

$$\neg \mathbf{B}(\mathbf{k}_\sigma, \mathbf{es}_\mathrm{k} = \mathbf{ms}_\mathrm{k}) \tag{73}$$

- In the case of readings (78) and (79), we get a similar effect, although here we mix the epistemic models of the speaker and Kim: one of the referring expressions is interpreted from the speaker's perspective while the other is again interpreted from Kim's perspective.
- For these two readings we obtain respectively the truth conditions in (74) and (75).

$$\neg \mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{ms}_\mathrm{k}) \tag{74}$$
$$\neg \mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{es}_\mathrm{k}) \tag{75}$$

- Finally for reading (80) we get the contradictory reading that Kim does not believe that Venus is Venus, as both referring expressions are evaluated using the speaker's perspective index.

$$\neg \mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{v}_\sigma) \tag{76}$$

- The different contexts for the interpretation of referring expressions are completely determined by the order in which we evaluate monadic resources.
- This means that, just by looking at the linear order of the lambda term, we can check whether a referring expression is evaluated inside the scope of a potentially perspective-changing operator such as *believe*, or if it is interpreted using the standard/speaker's interpretation.
- Notice that, given our internalist assumption about the nature of the model (see Asudeh and Giorgolo 2015), our analysis of a sentence like (72) does not specify what the actual case is with respect to the mind-external reality of any of the readings.

[14] The system generates six possible readings, as there are two possible orders of evaluation for the meaning of *Hesperus* and *Phosphorus* when they are both outside or inside the scope of *believe*. However, for our specific monad we have the following equality if x does not appear free in n and y does not appear free in m:

$$m \star \lambda x.n \star \lambda y.p = n \star \lambda y.m \star \lambda x.p \tag{71}$$

This captures the intuition that the interpretation value of independent expressions does not depend on the order of evaluation.

[15] The number of positive monadic types is irrelevant as any "stack" of monadic layers can be compressed into a single layer. This in fact corresponds to an alternative but equivalent definition of a monad, where the 'bind' operation ($\star$) is replaced by a so called 'join' operation (μ) that compresses two monadic layers into a single one.

$$\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Hesperus}\rrbracket \star \lambda x.\,\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{77}$$

$$\llbracket\text{Hesperus}\rrbracket \star \lambda x.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{78}$$

$$\llbracket\text{Phosphorus}\rrbracket \star \lambda x.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Hesperus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(y)(x)))(\llbracket\text{Kim}\rrbracket))) \tag{79}$$

$$\llbracket\text{Hesperus}\rrbracket \star \lambda x.\,\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{80}$$

Figure 1: Non-equivalent readings for *Kim doesn't believe Hesperus is Phosphorus*.

- Our system is based on the idea that the lexicon of a speaker is connected to her model of reality.
- The speaker's model, which is not necessarily representationally correct, also represents information that the speaker knows about the knowledge of other language users.
- For instance, in the case of the satisfiable readings for sentence (72), Kim's model will contain different axioms regarding the identities of the celestial bodies than the model of the speaker.
- In the scenario under consideration, the speaker knows facts about the world that Kim does not.
- Kim's mental model is not a completely accurate representation of reality, because Kim is unaware of an identity that should hold.
- But it is equally possible for the speaker's model to not adhere to reality.
- Before the discovery that Hesperus and Phosphorus are the same planet, a sentence like *Lysippus falsely believes that Hesperus is Phosphorus* would have been considered true.[16]
- If we consider a case like sentence (63), repeated in (81), we ought to get only a contradictory reading as there is no intuitively non-contradictory reading of the sentence (in the absence of focal stress on the second occurrence of *punch* or *Spider-Man*).

  (81) # Dr. Octopus punched Spider-Man but he didn't punch Spider-Man.

- Our analysis produces a single reading that indeed corresponds to a contradictory interpretation:

$$\begin{gathered}\llbracket\text{Spider-Man}\rrbracket \star \lambda x.\,\llbracket\text{Spider-Man}\rrbracket \star \\ \lambda y.\eta(\llbracket\text{but}\rrbracket\,(\llbracket\text{punch}\rrbracket\,(\llbracket\text{Dr. Octopus}\rrbracket)(x))(\llbracket\text{not}\rrbracket\,(\llbracket\text{punch}\rrbracket\,(\llbracket\text{Dr. Octopus}\rrbracket)(y))))\end{gathered} \tag{82}$$

- The verb *punch* is not a verb that can change the interpretation perspective and therefore the potentially controversial name *Spider-Man* is interpreted in both instances using the speaker's perspective index.
- The result is unsatisfiable truth conditions, as expected:

$$\mathbf{punch}(\mathbf{o}_\sigma, \mathbf{pp}_\sigma) \land \neg\mathbf{punch}(\mathbf{o}_\sigma, \mathbf{pp}_\sigma) \tag{83}$$

- In contrast a verb like *love* is defined in the lexicon in Table 5 as possibly changing the interpretation perspective about its object to that of its subject.
- Therefore in the case of a sentence like (64), repeated in (84), we expect one reading where the potentially contentious name *Spider-Man* is interpreted according to the subject of *love*, Mary Jane.

  (84) Mary Jane loves Peter Parker but she doesn't love Spider-Man.

[16]We operate under the assumption that the adverb *falsely* presupposes that the complement of the modified doxastic verb is false for the speaker of the sentence.

- This is in fact the result we obtain. Figure 2 reports the two readings that our framework generates for (84).
- Reading (87), corresponds to the non-contradictory interpretation of sentence (84), where Spider-Man is interpreted according to Mary Jane's perspective and therefore is assigned an entity different from Peter Parker:

$$\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \wedge \neg\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{sm}_{\mathtt{mj}}) \tag{85}$$

- Reading (88) instead generates unsatisfiable truth conditions, as Spider-Man is identified with Peter Parker according to the speaker's interpretation:

$$\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \wedge \neg\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \tag{86}$$

$$\begin{aligned}\eta(&[\![\text{but}]\!]\ ([\![\text{love}]\!]\ (\eta([\![\text{Peter Parker}]\!]))([\![\text{Mary Jane}]\!])) \\ &([\![\text{not}]\!]\ ([\![\text{love}]\!]\ ([\![\text{Spider-Man}]\!])([\![\text{Mary Jane}]\!]))))\end{aligned} \tag{87}$$

$$\begin{aligned}[\![\text{Spider-Man}]\!] \star \lambda x.\eta(&[\![\text{but}]\!]\ ([\![\text{love}]\!]\ (\eta([\![\text{Peter Parker}]\!]))([\![\text{Mary Jane}]\!])) \\ &([\![\text{not}]\!]\ ([\![\text{love}]\!]\ (\eta(x))([\![\text{Mary Jane}]\!]))))\end{aligned} \tag{88}$$

Figure 2: Non-equivalent readings for *Mary Jane loves Peter Parker but she doesn't love Spider-Man.*

- Our last example, the Capgras example (65), repeated here as (89), is particularly interesting as the embedded clause is just a simple identity statement with two tokens of the same name.
- We are not aware of formal analysis of this kind of example in the literature, the closest being the Hecdnett example in Castañeda [1989].
- The non-contradictory reading that this sentence has seems to be connected specifically to two different interpretations of the same name, *Sandy*, both syntactically embedded under the propositional attitude verb *believe*.

  (89) Kim doesn't believe Sandy is Sandy.

- Our system generates three non-equivalent readings, reported here in Figure 3.[17]
- Reading (93) and (94) are two contradictory readings of the sentence.
- In the first case, both instances of the name *Sandy* are interpreted from the subject's perspective and therefore a lack of belief in a tautology is attributed to Kim.
- In the second case, both instances of the name *Sandy* are interpreted from the speaker's perspective, again resulting in an assertion that Kim does not believe a tautology.
- In contrast the reading in (95) corresponds to the interpretation that assigns two different referents to the two instances of the name *Sandy*, producing the truth conditions in (90) which are satisfiable in a suitable model.

$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{s}_\sigma = \mathbf{imp}_{\mathtt{k}}) \tag{90}$$

- We use $\mathbf{imp}_{\mathtt{k}}$ as the speaker's representation of the "impostor" that Kim thinks has taken the place of Sandy.

[17] Again the system generates six non-equivalent readings (see footnote 14), which are further reduced in this case as we have the same linguistic term appearing twice and combined with a commutative predicate (*is*).

- The analysis of the Aslan/Jesus example (55), repeated in (91), is equivalent; the non-contradictory reading is shown in (92).

(91) Reza doesn't believe Jesus is Jesus.

$$\neg \mathbf{B}(\mathbf{r}_\sigma, \mathbf{j}_\sigma = \mathbf{j}_\mathbf{r}) \tag{92}$$

- There are again three non-equivalent readings, including the one above, which are just those in Figure 3, with ⟦Sandy⟧ replaced by ⟦Jesus⟧ and ⟦Kim⟧ replaced by ⟦Reza⟧.

$$\eta(\llbracket \text{not} \rrbracket\,(\llbracket \text{believe} \rrbracket\,(\llbracket \text{Sandy} \rrbracket \star \lambda x.\,\llbracket \text{Sandy} \rrbracket \star \lambda y.\eta(\llbracket \text{is} \rrbracket\,(x)(y)))(\llbracket \text{Kim} \rrbracket))) \tag{93}$$

$$\llbracket \text{Sandy} \rrbracket \star \lambda x.\,\llbracket \text{Sandy} \rrbracket \star \lambda y.\eta(\llbracket \text{not} \rrbracket\,(\llbracket \text{believe} \rrbracket\,(\eta(\llbracket \text{is} \rrbracket\,(x)(y)))(\llbracket \text{Kim} \rrbracket))) \tag{94}$$

$$\llbracket \text{Sandy} \rrbracket \star \lambda x.\eta(\llbracket \text{not} \rrbracket\,(\llbracket \text{believe} \rrbracket\,(\llbracket \text{Sandy} \rrbracket \star \lambda y.\eta(\llbracket \text{is} \rrbracket\,(x)(y)))(\llbracket \text{Kim} \rrbracket))) \tag{95}$$

Figure 3: Non-equivalent readings for *Kim doesn't believe Sandy is Sandy.*

References

Scott AnderBois, Adrian Brasoveanu, and Robert Henderson. At-issue proposals and appositive impositions in discourse. *Journal of Semantics*, 32(1):93–138, 2015. doi: 10.1093/jos/fft014. URL `http://jos.oxfordjournals.org/content/32/1/93.abstract`.

Ash Asudeh and Gianluca Giorgolo. Perspectives. Ms., University of Oxford and Carleton University, 2015.

Héctor-Neri Castañeda. *Thinking, Language, and Experience*. University of Minnesota Press, Minneapolis, 1989.

Simon Charlow. *On the Semantics of Exceptional Scope*. PhD thesis, New York University, 2014.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.

Noam Chomsky. *Knowledge of Language: Its nature, origin, and use*. Praeger, New York, NY, 1986.

Noam Chomsky. *New Horizons in the Study of Language and Mind*. Cambridge University Press, Cambridge, 2000.

Gianluca Giorgolo and Ash Asudeh. Multidimensional semantics with unidimensional glue logic. In Miriam Butt and Tracy Halloway King, editors, *Proceedings of the LFG11 Conference*, pages 236–256. CSLI Publications, 2011.

Gianluca Giorgolo and Ash Asudeh. Missing resources in a resource-sensitive semantics. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG12 Conference*, pages 219–239. CSLI Publications, Stanford, CA, 2012a.

Gianluca Giorgolo and Ash Asudeh. $\langle M, \eta, \star \rangle$ Monads for conventional implicatures. In Ana Aguilar Guevara, Anna Chernilovskaya, and Rick Nouwen, editors, *Proceedings of Sinn und Bedeutung 16*, volume 1, pages 265–278. MIT Working Papers in Linguistics, 2012b.

Gianluca Giorgolo and Ash Asudeh. Monads as a solution for generalized opacity. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 19–27, Gothenburg, 2014a. Association for Computational Linguistics.

Gianluca Giorgolo and Ash Asudeh. One semiring to rule them all. In *Proceedings of the 36th Annual Cognitive Science Society Conference*, pages 116–121, 2014b.

Gianluca Giorgolo and Christina Unger. Coreference without discourse referents: A non-representational DRT-like discourse semantics. In B. Plank, T. Kim Sang, and T. Van de Cruys, editors, *Computational Linguistics in the Netherlands 2009*, number 14 in LOT Occasional Series, pages 69–81. LOT, 2009.

H. Paul Grice. *Studies in the Way of Words*. Harvard University Press, Cambridge, MA, 1989.

Daniel Gutzmann. *Use-Conditional Meaning: Studies in Multidimensional Semantics*. Oxford University Press, Oxford, 2015.

Irene Heim and Angelika Kratzer. *Semantics in Generative Grammar*. Blackwell, Oxford, 1998.

Ralph Hertwig and Gerd Gigerenzer. The "conjunction fallacy" revisited: How intelligent inferences look like reasoning errors. *Journal of Behavioral Decision Making*, 12:275–305, 1999.

Ray Jackendoff. *Semantics and Cognition*. MIT Press, Cambridge, MA, 1983.

Ray Jackendoff. *The Architecture of the Language Faculty*. MIT Press, Cambridge, MA, 1997.

Ray Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, Oxford, 2002.

Ray Jackendoff. *Language, Consciousness, Culture: Essays on Mental Structure*. MIT Press, Cambridge, MA, 2007.

Stephen C. Levinson. *Pragmatics*. Cambridge University Press, Cambridge, 1983.

Eric McCready. Varieties of conventional implicature. *Semantics and Pragmatics*, 3(8):1–57, July 2010. doi: 10.3765/sp.3.8.

Eugenio Moggi. Computational lambda-calculus and monads. In *LICS*, pages 14–23. IEEE Computer Society Press, 1989. ISBN 0-8186-1954-6.

Eugenio Moggi. An abstract view of programming languages. Technical report, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, Edinburgh, 1990.

Christopher Potts. *The Logic of Conventional Implicatures*. Oxford University Press, Oxford, 2005.

Jennifer Saul. *Simple Sentences, Substitution, and Intuitions*. Oxford University Press, Oxford, 2007.

Jennifer M. Saul. Substitution and simple sentences. *Analysis*, 57(2):102–108, 1997.

Chung-chieh Shan. Monads for natural language semantics. In Kristina Striegnitz, editor, *Proceedings of the ESSLLI-2001 Student Session*, pages 285–298. 13th European Summer School in Logic, Language and Information, 2001.

Amos Tversky and Daniel Kahneman. Extensional Versus Intuitive Reasoning: The Conjunction Fallacy in Probability Judgment. *Psychological Review*, 90(4):293–315, 1983.

Christina Unger. Dynamic semantics as monadic computation. In Manabu Okumura, Daisuke Bekki, and Ken Satoh, editors, *New Frontiers in Artificial Intelligence — JSAI-isAI 2011*, pages 68–81, 2011.

Philip Wadler. In *Proceedings of the 19th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, New York, NY, USA.

Philip Wadler. Comprehending monads. *Mathematical Structures in Computer Science*, 2:61–78, 1992.

Philip Wadler. Monads and composable continuations. *Lisp and Symbolic Computation*, 7(1):39–56, 1994.

Philip Wadler. Monads for functional programming. In Johan Jeuring and Erik Meijer, editors, *Advanced Functional Programming*, pages 24–52. Springer, Berlin, 1995.

Thomas Ede Zimmermann. What's in two names? *Journal of Semantics*, 22:53–96, 2005.