Natural Language Semantics with Enriched Meanings

Gianluca Giorgolo and Ash Asudeh

August 18, 2015

Contents

Lecture 1

Multidimensionality, Perspectives, Uncertainty, and Monads

1.1 Introduction

This course is about using a concept from category theory and functional programming — *monads* — to model certain murkier aspects of natural language meaning, following ideas that came to us through Shan [2001]. 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, but also to lay out the formal tools and to provide you with hands-on experience, through the prover that Gianluca has developed, to take the program further, if you find it interesting.

In this first lecture, we lay out some of the basics: the standard bifurcation of meaning into semantics and pragmatics — including problems at the borders, the notion of enriched meanings, and some particular phenomena that seem to flicker between the realms of meaning — but, mostly, some category theory.

1.2 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 1.1.

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

Table 1.1 Conventions vs. truth conditions [Gutzmann, 2015, 5]

- The cells marked [+conventional,+truth-conditional] and [−conventional,−truth-conditional] are respectively "clear-cut" cases of semantics and pragmatics.
- The other two cells are the borderlands between semantics & pragmatics, where a lot of the interesting action is.
- It is the borderlands that this course is 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]:

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

  (1.2) 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.[1]

- **Pragmatic enrichment**

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

  (1.3) 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.

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

[1] This is not to say that the fact that they are synonyms is necessarily directly part of their conventional meaning. That depends on your theory of lexical semantics.

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. 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).

  (1.5) This dog howled the whole night.

  (1.6) This cur howled the whole night.

  (1.7) 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.[2]

  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:

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

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

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

1.3 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, 1992a,b, 1994, 1995, Shan, 2001, Giorgolo and Unger, 2009, Giorgolo and Asudeh, 2011, 2012a,b, 2014a,b, Asudeh and Giorgolo, 2015, Charlow, 2014].

- What are some intuitive advantages of our approach?

  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.

1.4 The Phenomena[3]

1.4.1 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.

(1.10) 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.

(1.11) 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].

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

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

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

(1.15) 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:

(1.16) They are Krauts.

(1.17) Your cur bit me.

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

1.4.2 Perspectives: Reference and substitution

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

  (1.18) Hesperus is Phosphorus. *True*

  (1.19) Kim believes that Hesperus is a planet. $\not\equiv$

  (1.20) Kim believes that Phosphorus is a planet.

  (1.21) 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].

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

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

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

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

  (1.26) 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:

  (1.27) #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]

  (1.28) 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]

  (1.29) 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].

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

	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 1.2 Overview of substitution puzzles.

1.4.3 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:

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

- 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.

  (1.31) 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 Gricean reasoning — using two related *semirings* and monadic machinery to compose and store probabilistic measures associated with semantic types.

1.5 Course Roadmap

Day 1 Introduction to Key Formal Concepts

Day 2 Multidimensionality: Conventional Implicatures

Day 3 Perspectives: Substitution Puzzles

Day 4 Uncertainty: Conjunction Fallacies

Day 5 Interactions, Connections & Wrap-up

1.6 A Little Bit of Category Theory

What follows does not attempt to be an introduction to category theory. There are plently of good introductions about the topic (see for example Barr and Wells [1990], Pierce [1991]). Here we are just trying to give an impressionistic idea of category theory, introducing some core ideas that will be used in the rest of these notes.

There is no magic in what we will be doing, and we will not really create any new kind of semantic theory. Instead, we will look at how certain ways in which standard compositional semantics has been bent to fit difficult data can be described in the language of category theory and how these descriptions seem to point to a common type of construction, the monad.

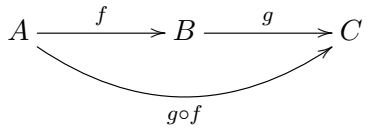
1.6.1 Categories

Categories pop up more or less everywhere, so it is not easy to give an intuitive definition of what a category is (vs. what it is not). We can think of a category as a way to describe some kind of complex ensamble of objects by specifying in which ways these objects interact. This is more or less the perspective we will take here as far as linguistic meanings are concerned: our objects will be different kinds of meanings (entities, proposition, predicates, relations, quantifiers...), but instead of focusing on what they are, we focus on how they interact, and in particular how by composing them we can construct more complex meanings. In any case here is the formal definition: a category $\mathbf{C}$ is a collection $\mathbf{C_1}$ of *objects* together with a collection $\mathbf{C_2}$ of *arrows* or *morphisms* satisfying the following conditions:

1. For each arrow f we have two functions $dom : \mathbf{C_2} \to \mathbf{C_1}$ and $cod : \mathbf{C_2} \to \mathbf{C_1}$, such that $dom(f)$ is the source object of the arrow and $cod(f)$ the target object of f. If $dom(f) = A$ and $cod(f) = B$ we write compactly $f : A \to B$ and draw the arrow as follows:

$$A \xrightarrow{\quad f \quad} B$$

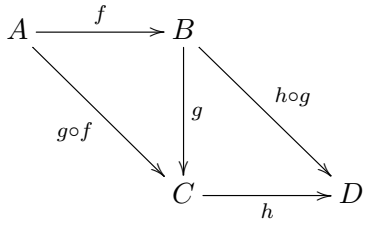
2. For all arrows f and g, if $cod(f) = dom(g)$ then there is an arrow $g \circ f$ with source $dom(f)$ and target $cod(g)$, graphically:



   We call $\circ$ the *arrow composition* operation.

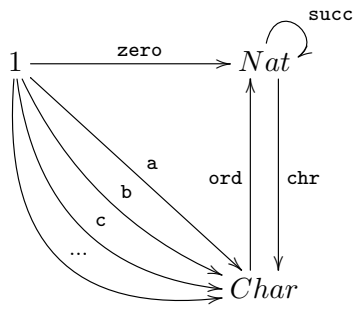
3. For each object A there is an arrow id_A with source and target A, and such that for all arrows $f : A \to B$ we have that $f \circ id_A = f$ and for all arrows $g : B \to A$ we have that $id_A \circ g = g$. id_A is called the identity arrow for object A.[6]

4. Arrow composition is associative, i.e. for all (composable) arrows f, g, h we have that $h \circ (g \circ f) = (h \circ g) \circ f$, graphically:



A note on the terminology: in the case of a diagram like the one above we will say that the associativity of arrow composition is verified if the diagram *commutes*, which means that all paths between any two objects have the same result. This means different things in different categories, for instance is the arrows are functions and we have different ways to go from one set to another (via the composition of different functions), the resulting composed functions should be equal (i.e. give the same results for the same arguments).

Example Consider the following categorical model of a very simple programming language with just three types, a type 1 with a single value, the type Nat of natural numbers and the type $Char$ of characters:



[6]Identities are unique. Suppose there is an arrow $j : A \to A$ such that for all $f : A \to B$ and $g : B \to A$ we have that $f \circ j = f$ and $j \circ g = g$ then $id_A = id_A \circ j = j$.

As we said the objects of this category are the types of values in this mini language, while arrows represent programs, that take as input values whose type is the source of the arrow and that return values whose type is the target object of the arrow. For instance `ord` is the program that returns for a given character its numeric value in a predefined encoding.

1 is called a *terminal object*. The defining property of a terminal object is that for each object X in the category there is exactly one arrow between X and the terminal object. In our case, given that 1 has a single inhabitant, every program must map all input values to that same value. Using arrows *from* the object 1 to any type we can identify the inhabitants of the type. This is because for each inhabitant of the target object there is exactly one program that maps the single value in 1 to that inhabitant. For instance, there is one arrow between 1 and $Char$ for each character in the encoding. Similarly there is one arrow between 1 and Nat for each natural number: one of them is `zero` which picks the number zero, the others are equal to the iterated composition of the `succ` arrow with `zero`, for instance the arrow $\texttt{succ} \circ \texttt{succ} \circ \texttt{zero}$ identifies the number two.

In general, in this simple category programs are described by composing arrows. For instance, we can describe the program that given a character returns the next one in the encoding using the arrows in the diagram: $\texttt{chr} \circ \texttt{succ} \circ \texttt{ord}$.

Notice that the language described by this model is extremely limited. Given this definition we cannot for instance describe programs that take more than a single input value, and in general we don't have a way to talk about functions as objects and the result of applying them to their argument (this in turn means we cannot describe recursive / iterative processes).

We extend our programming language with a new type $Nat \times Nat$ representing pairs of natural numbers (see figure 1.1). We also introduce two new arrows / programs: `minus` that maps a pair of natural numbers to the result of subracting the second number of the pair from the first one,[7] and the arrow $\langle \_, 32 \rangle$ which maps a natural number n to the the pair $\langle n, 32 \rangle$. Assuming the standard ASCII encoding as the result of the `ord` map, where uppercase letters are mapped in alphabetic order to the range $[65, 90]$, while lowercase letters are mapped to the range $[97, 122]$, we can use the defined arrows to write the program `to_upper` that sends a lower case character to its uppercase version (we don't care about the results we get when the program is applied to non lowercase character): $\texttt{chr} \circ \texttt{minus} \circ \langle \_, 32 \rangle \circ \texttt{ord}$.

Category of linguistic meanings The category of linguistic meanings is a bigger version of the category for the mini programming language we saw before, with additional gadgets that allows us to relate it to the language we are used to use when doing semantics, the *simply-typed lambda calculus*. The first two gadgets are *product* and *exponential* objects.

For our purposes it is sufficient to say that *products* correspond to the *pairing* of two objects that can be taken apart and recomposed, meaning that we can operate on the

[7] We assume that `minus` is defined for all pairs of naturals but that it maps each pair whose second number is greater than the first one to the "incorrect" result zero.

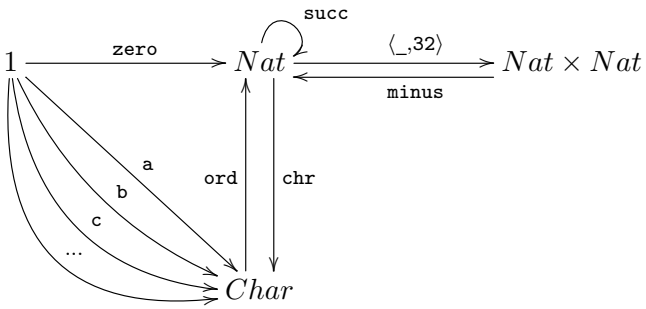


Figure 1.1 Extended micro language.

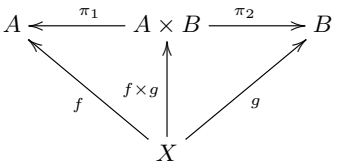
joined objects independently of each other.[8] We will write the object that corresponds to the pairing of an object A and an object B as $A \times B$. So for instance, given an object t representing the type of truth-values, a pair of truth values will be living in the object $t \times t$. As you may expect, for each product object we have projection functions that π_1 and π_2 that map the product to its two factors.

Exponential objects instead represent collection of functions. In our category they correspond to the collection of arrows between two objects. So for instance, the object B^A will be the collection of arrows that go from A to B. In our case the object t^e will be the collection of functions between the object e (the domain of the discourse) and t (the truth-values). To be more in line with the way we are normally used to write functional types, in what follows we will write t^e as $e \to t$.

Finally we will assume that there is an arrow $eval : (A \to B \times A) \to B$ that behaves as expected and performs the application of an element of $A \to B$ to one of A, returning the result in B.[9]

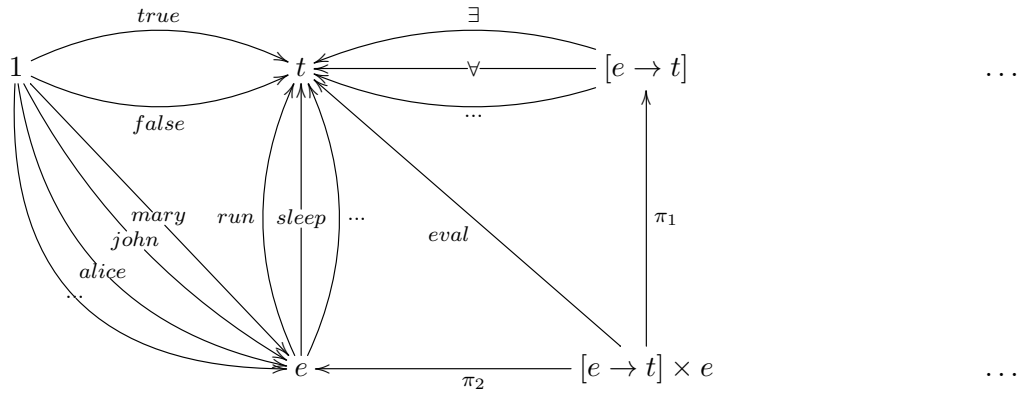
Graphically we can represent a tiny part of the category of linguistic meanings as

[8] This property corresponds to the universal property of product objects that says that given a product $A \times B$, then for any object X and arrows $f : X \to A$ and $g : X \to B$ there is a unique arrow $f \times g : X \to A \times B$ for which the following diagram commutes:



[9] This is the morphism that defines the universal property of exponentials.

follows:



1.6.2 Functors

Functors are maps between categories. They map the objects of the starting category to objects of the target one, and similarly for arrows. Functors though have a contract to fullfill: the mapping must respect the rules of arrow composition and the notion of identity arrows.

But functors can also define a mapping inside a single category. This is the way we are going to use them, in particular to map certain classes of meaning into other classes,

Given two categories **C** and **D**, a functor $F : \mathbf{C} \to \mathbf{D}$ is composed of two mappings, F_1 and F_2: F_1 maps **C**-objects into **D**-objects, while F_2 maps **C**-arrows $f : A \to B$ to **D**-arrows $F_2(f) : F_1(A) \to F_1(B)$, such that for all **C**-objects and composable **C**-arrows f and g the following holds:

1. $F_2(id_A) = id_{F_1(A)}$
2. $F_2(g \circ f) = F_2(g) \circ F_2(f)$

The two categories **C** and **D** can also be the same category. In this case we talk about an *endofunctor*. An endofunctor can be used to map some part of our object space to another part. This is the kind of functors we will be using as we will be working inside the category of linguistic meanings.

Example A computer science inspired example of an endofunctor is $List$ which maps a set A to the set $List_1(A)$ of lists over elements of A. Suppose that our A is the set $\{a, b, c\}$, then:

$$List_1(\{a, b, c\}) = \{\varepsilon, [a], [b], [c], [a, a], [a, b], [a, c], [b, a], [b, b], [b, c], [c, a], [c, b], [c, c], ...\}$$

and which sends each arrow/function $f : A \to B$ to a function $List_2(f) : List_1(A) \to List_1(B)$ that applies f to every element of the input list. Now suppose that A is as given above and B is the set $\{0, 1\}$, then we have:

$$f = \{a \mapsto 0, b \mapsto 1, c \mapsto 1\}$$

$$List_2(f) = \{\varepsilon \mapsto \varepsilon, [a] \mapsto [0], [b] \mapsto [1], [c] \mapsto [1], [a, a] \mapsto [0, 0], [a, b] \mapsto [0, 1], [a, c] \mapsto [0, 1], ...\}$$

Example Another example from computer science of an endofunctor is the functor $Annotate$ that send a set A to the cartesian product of the set with the element set of a monoid M, $A \times M$.

A monoid is an algebraic structure composed by a set M together with a binary associative operation, the multiplication, $\cdot : M \times M \to M$ and a distinguished element 1, the unit, such that for all $x \in M$ we have that $x \cdot 1 = 1 \cdot x = x$.

The functor under discussion maps set functions so that they operate only on the first component of the products in $A \times M$:

$$f = \{a \mapsto 0, b \mapsto 1, c \mapsto 1\}$$

$$Annotate_2(f) = \{\langle a, \varepsilon\rangle \mapsto \langle 0, \varepsilon\rangle, \langle a, [a, b]\rangle \mapsto \langle 0, [a, b]\rangle, \langle b, [a]\rangle \mapsto \langle 1, [a]\rangle, ...\}$$

Categories and functors form a category: objects are categories and arrows are functors. For each category $\mathbf{C}$ there is an identity functor $Id : \mathbf{C} \to \mathbf{C}$, mapping objects and arrows to themselves, and functors can be composed in the obvious way.

1.6.3 Natural Transformations

The idea of a natural transformation is a bit more hard to describe intuitively. If functors describe ways of mapping different categories in such a way that the structure of arrow composition is preserved, natural transformation describe ways of mapping functors into functors, in a way that is "natural".

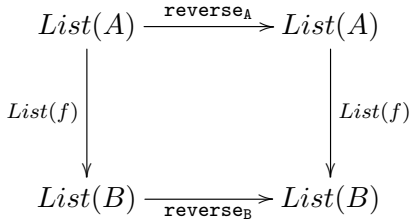
Given two categories $\mathbf{C}$ and $\mathbf{D}$ and two functors $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{C} \to \mathbf{D}$, a natural transformation $\eta : F \to G$ is a function that assigns to every $\mathbf{C}$-object a $\mathbf{D}$-arrow $\eta_A : F(A) \to G(A)$ such that for every $\mathbf{C}$-arrow $f : A \to B$ the following diagram commutes (that is all paths lead to the same result):

$$\begin{array}{ccc} F(A) & \xrightarrow{\eta_A} & G(A) \\ {\scriptstyle F(f)}\downarrow & & \downarrow{\scriptstyle G(f)} \\ F(B) & \xrightarrow[\eta_B]{} & G(B) \end{array}$$

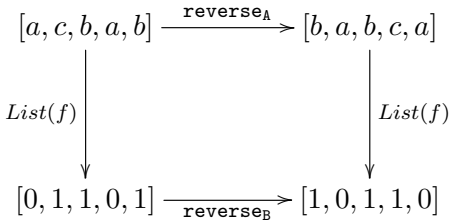
Example Probably the most intuitive examples of natural transformations come from computer science.

The function `reverse` that reverses the order of a list is a natural transformation. `reverse` is a polymorphic function, we can apply it to a list of any type of objects. `reverse` is a natural transformation between $List$ and itself. To prove that this is the case we have just to prove that reversing the order of a list and then applying a function f to all its elements is equivalent to first applying f to the elements of the original list and then

reversing the order of the result, or diagrammatically:

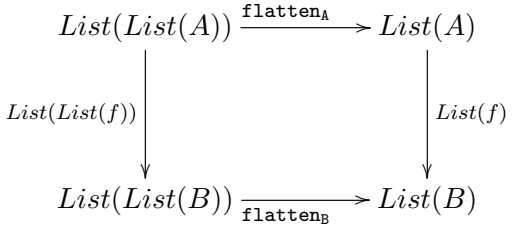


To look at this in a more concrete way consider the f function we defined in section 1.6.2, it is easy to see how the diagram commute for a specific example:

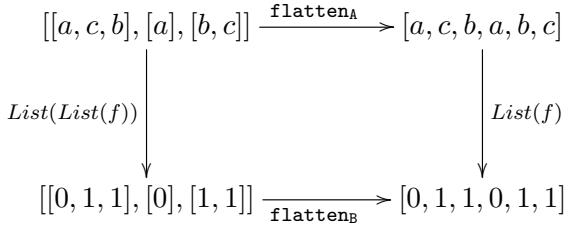


Another (although probably not terribly useful) example of a natural transformation between the functor $List$ and itself is the function `cycle` that takes a list and returns an infinite list composed of an infinite number of copies of the same list (on an infinite list `cycle` is just the identity function, which by the way is a natural transformation). Saying that this is a natural transformation means that it doesn't matter if we first create an infinite version of a list of objects and then apply a certain function to each of them, or rather if we first apply the function to each element of the starting list and then create the infinite version.

A probably more interesting example, is a function `flatten` that "flattens" a list of lists of objects into a single-layered list, e.g. $\texttt{flatten}([[a, c, b], [a], [b, c]]) = [a, c, b, a, b, c]$. `flatten` is a natural transformation between the functor $List \circ List$ and $List$. Again, when we say that `flatten` is a natural transformation it means that it doesn not matter if we first flatten a list of lists and then apply a function to all its element, or if we first apply the function to all elements of all lists of the initial list and then flatten the result:



So for example:



1.6.4 Monads

Monads are a particular type of endofunctors that are equipped with two natural transformations that encode the notions of *embedding* and *joining / composing / combining* (this intuition is taken from Kuhnle [2013]). This probably not very illuminating so let's try to motivate them with an example. The following discussion borrows massively from Spivak [2013], which, by the way, is a very good introduction to category theory for non-mathematicians.

Remember that a fundamental component of a category is the operation that allows us to compose two arrows. The kind of arrows we have dealt with so far have been *functions*[10], but what about *partial functions*? It turns out that we can define quite easily a category whose objects are sets and whose arrows are partial functions (considering the identities as a special kind of partial functions). By composing to partial functions $f : A \to B$ and $g : B \to C$ we get a third partial function $g \circ f$ from A to C that is defined only for those elements of A for which f is defined and whose image in C are mapped by g to an element of C.

But now when we write $g \circ f$ we have to keep in mind that this is not ordinary function composition if we are dealing with partial functions. We could just consider all functions partial. But composition of partial functions is kind of complicated (we used a lot of words to describe it). There is actually a simple way to represent a partial function as a total one, by extending its codomain with a new distinguished element, say ♣, and by sending each element of the domain for which the original function is not defined to ♣. So instead of having a *partial* function $f : A \to B$ we will have a total function $f' : A \to B \sqcup \{\clubsuit\}$ ($\sqcup$ is the disjoint union of two sets). Notice also that we can also modify total functions to fit this shape, simply by embedding the codomain of the function into its union with $\{\clubsuit\}$:

$$\eta : B \to B \sqcup \{\clubsuit\} \tag{1.32}$$

In this way we do not need anymore to talk about a different kind of arrow composition. But we have another problem now: the source and target objects of partial functions that we could compose before do not match anymore. There is actually a way to fix also this minor setback.

Say that we want to compose one of our new functions $f : A \to B \cup \{\clubsuit\}$ with $g : B \to C \cup \{\clubsuit\}$. The first thing to do is to create a new function

$$g' : B \sqcup \{\clubsuit\} \to C \sqcup \{\clubsuit\} \sqcup \{\clubsuit\} \tag{1.33}$$

[10] From our discussion it may have seemed that arrows are always functions but this is not the case.

that behaves exactly like g except that it sends ♣ to another copy of ♣. We then have a composite arrow $g' \circ f : A \to C \sqcup \{\clubsuit\} \sqcup \{\clubsuit\}$, which we can compose with

$$\mu : C \sqcup \{\clubsuit\} \sqcup \{\clubsuit\} \to C \sqcup \{\clubsuit\} \tag{1.34}$$

to obtain a composite arrow $\mu \circ g' \circ f : A \to C \sqcup \{\clubsuit\}$ with the desired source and target.

It turns out that all the seemingly arbitrary kind of operations we have used to get a way to compose partial and total function in a uniform way are all instances of the categorical constructions we have introduced so far. First of all we have made use of the fact that the disjoint union of a set with our special element ♣ forms an endofunctor in the category that has sets as objects and functions as arrows: sets are mapped to their disjoint union with ♣ and arrows$A \to B$ are mapped to arrows $A \sqcup \{\clubsuit\} \to B \sqcup \{\clubsuit\}$ in the same way we did in the case of g' in (1.33) by simply extending the function so that it sends ♣ to ♣. We will call this functor $\cdot \sqcup \{\clubsuit\}$.

The embedding function, or better the family of functions,[11] defined in (1.32) is instead a natural transformation between the functor Id (the identity function that as expected maps objects and arrows to themselves) and our functor.

Finally the kind of function defined in (1.34) is also an instance of a natural transformation, this time between the functor $\cdot \sqcup \{\clubsuit\} \circ \cdot \sqcup \{\clubsuit\}$, obtained by composing the functor with itself, and $\cdot \sqcup \{\clubsuit\}$.

And this is all there is to a monad. And here are some more examples of monads:

Example Consider a set A and the set $List(A)$ of lists over A. We know that this is a functor (we can map a function from A to B to a function from $List(A)$ to $List(B)$). We can embed A into $List(A)$, by mapping each element x of A to the list of length 1 $[x]$. If we have a list of lists of elements of A, i.e. an element of $List(List(A))$ we can "lower" it to an element of $List(A)$ by concatenating the inner lists: $[[a, b, c], [b, c, a], [a, b]] \rightsquigarrow [a, b, c, b, c, a, a, b]$, with our friend `flatten`$i$.

Example Take a set A and a monoid M. We know that there is a functor that maps A to the cartesian product $A \times M$. There is also a way to embed A into $A \times M$, by pairing every element of A with the monoid unit 1. Also, if we apply two times the functor we get objects of type $(A \times M) \times M$, which we can lower to the type $A \times M$ by using the monoid multiplication: $\langle\langle x, m_1\rangle, m_2\rangle \rightsquigarrow \langle x, m_1 \cdot m_2\rangle$.

Example Another important functor is the one that maps a set A to the set of functions from some set I to A, $A \to (I \to A)$. A function from A to B is mapped by this functor to a function from $I \to A$ to $I \to B$ in this way:

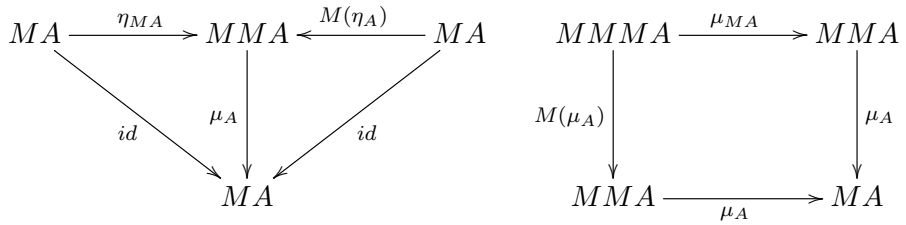
$$f_{A \to B} \mapsto \lambda g_{I \to A}.\lambda i_I.f(g(i))_B$$

[11] Remember that a natural transformation maps each object in the target category of the first functor to an *arrow* in the target category of the second functor, therefore we have different functions indexed by the source objects (in our case different linguistic types) of the transformation

There is a way in which we can embed the set A inside $I \to A$, simply by mapping each element of x to the function that always returns x. And we can lower a function of type $I \to (I \to A)$ to one of type $I \to A$ by simply passing two times the same argument:

$$f_{I \to (I \to A)} \mapsto \lambda i_I.(f(i))(i)_A$$

In general we say that a monad is a triple $\langle M, \eta, \mu \rangle$, where M is an endofunctor, η is a natural transformation from Id to M, and μ a natural transformation from M^2 (the composition of M with itself) to M. The natural transformations must make the following two (obscure) diagrams commute:



Intuitively we want η (the embedding) to be such that we can in some sense undo it via μ (the join) without any consequence.

In what follows we will use a different definition of a monad, where instead of the join transformation we have a different operation that we call *bind* and write as $\star$. Bind basically takes an arrow $A \to M(B)$ and maps it to an arrow $M(A) \to M(B)$ (we will use a slightly different type signature for $\star$, instead of $(A \to M(B)) \to M(A) \to M(B)$ we will use $M(A) \to (A \to M(B)) \to M(B)$). The advantage of using bind instead of join is that it works a bit like a sort of functional application (at least if you swap the order of its arguments) and will make the meaning terms a bit more understandable.

But the two ways of defining monads are completely equivalent, and in fact we can express bind in terms of join:

$$m \star f = \mu(M_2(f)(m))$$

or join in terms of bind:

$$\mu(m) = m \star id$$

According to this alternative definition a monad is a triple $\langle M, \eta, \star \rangle$, where M and η are the same objects as before, while $star$ is the operation we have described above. We require η and $\star$ to satisfy the following three laws:

$$m \star \eta = m \tag{1.35}$$

$$\eta(x) \star f = f(x) \tag{1.36}$$

$$(m \star f) \star g = m \star (\lambda x.f(x) \star g) \tag{1.37}$$

The first two laws say that η behaves as a kind of unit if we consider $\star$ to be some sort of multiplication, while the last law ensures that $\star$ is associative.

There is another interpretation that we can associate with this definition of monads. We can view monads as encoding *computation* that return a certain type of result. But in doing so they may have some side effects. So we read $M(A)$ as the type of computations that return results of type A but that have side effects M. Then η becomes a way to create a computation that returns a specific value *without* actually producing any real side effect (it's a unit). $\star$ is instead the way in which we compose computations. Its type looks like a sort of power functional application, and we understand $\star$ as basically running the computation that returns as result an object that is the input for the second argument of $\star$, applying the second argument to result and returning the resulting computation. At the same time $\star$ combines the side effects of the computations.

1.7 From a Theory of Categories to a Lambda Calculus to a Logic

It would be an interesting exercise to try to formalize natural language semantics using the language of category theory, but we would need to introduce a lot more of the theory and this is definitely beyond the scope of this short course. Instead we will use the computational lambda calculus introduced by Moggi [1989], which, for our purposes, is basically the lambda-calculus with the addition of a couple more types of expressions to deal with monads. The two additions are the unit η and the bind operation $\star$. At the level of types of our expressions, we will have a new type for monadic terms signaled by the use of $\Diamond$. When we will want to distinguish between different monadic types we will a subscript to it to indicate which monad we are talking about, e.g. $\Diamond_{ci}$.

Here is the definition of our computational lambda calculus:

Definition 1. We assume we have a set X of typed variables and a set C of type constants:

- a variable is a term of the specified type,
- so is a constant,
- if x is a variable of type τ and t is a term of type δ, then $\lambda x.t$ is a term of type $\tau \to \delta$,
- if t is a term of type $\tau \to \delta$ and u is a term of type τ, then $t(u)$ is a term of type δ,
- if t is a term of type τ, then $\eta(t)$ is a term of type $\Diamond\tau$ (for the relevant monad),
- if m is a term of type $\Diamond\tau$, x a variable of type τ and n a term of type $\Diamond\delta$, then $m \star \lambda x.n$ is a term of type $\Diamond\delta$ (the scope of $\star$ extends to the right as far as possible, so if we have $m \star \lambda x.n \star \lambda y.p$ it is equivalent to $m \star (\lambda x.n \star (\lambda y.p))$).

The final step to actually use this language for doing semantics is to attach it to a logical calculus, so that the calculus can construct the complex terms that correspond to the meaning of complex expressions for us starting from the atoms of the language. We will use the beautiful calculus introduced in Benton et al. [1998] and reproduced here in a sequent calculus format in figure 1.2. The only difference here is that our calculus is not only intuitionistic but also linear, meaning that that we do not have rules like *contraction* or *weakening*.

$$\frac{}{x : A \vdash x : A}\, id \qquad \frac{\Gamma \vdash B \qquad B, \Delta \vdash C}{\Gamma, \Delta \vdash C}\, Cut$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \multimap B} \multimap R \qquad \frac{\Delta \vdash t : A \qquad \Gamma, x : B \vdash u : C}{\Gamma, \Delta, y : A \multimap B \vdash u[y(t)/x] : C} \multimap L$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u \rangle : A \otimes B} \otimes R \qquad \frac{\Gamma, x : A, y : B \vdash t : C}{\Gamma, \langle u, v \rangle : A \otimes B \vdash t[x/u, y/v] : C} \otimes L$$

$$\frac{\Gamma \vdash x : A}{\Gamma \vdash \eta(x) : \Diamond A} \Diamond R \qquad \frac{\Gamma, x : A \vdash t : \Diamond B}{\Gamma, y : \Diamond A \vdash y \star \lambda x.t : \Diamond B} \Diamond L$$

Figure 1.2 Sequent calculus for a fragment of multiplicative linear logic with a monadic modality.

The axiom rule, the Cut rule and the rules for the introduction to the right and the left of the linear implication $\multimap$ and the tensor $\otimes$ are familiar from standard linear logic, but the rules for the monadic connective $\Diamond$ deserve some comment. The rule for introduction to the right of $\Diamond$ when read from top to bottom basically states that if we have a derivation that proves that from a set of resources/hypotheses Γ we can derive a resource/proposition A, then we are also able to derive from the same set of resources/hypotheses the same resource/proposition "encapsulated" into a monadic layer. If we interpret $\Diamond$ as representing an operation that allows us to lift a value into a new space of values, the rule allows us to lift a value into this new space. If we interpret the monadic connective in terms of computations, the rule states that we can create a computation that has as a result the resource that we started with.

The nature of this computation is perhaps better understood if we look at the lambda term associated with the rule. The new computation is produced by wrapping the original result with the unit η. Remember that the unit acts as a lifting operator that actually does not do anything besides taking the original value to a different value space. In this sense the right-hand rule for $\Diamond$ is safe as it does not introduce any additional information in the system.[12]

The rule for introduction of $\Diamond$ to the left is slightly more complicated. First of all notice that, despite being a rule governing the behaviour of a connective on the left-hand side of the turnstyle, the rule imposes some restrictions on what is on the right-hand side of the sequent, namely that the resulting resource/proposition is a monadic one (but notice that with the right-hand rule for $\Diamond$ we can always construct a "dummy" monadic resource). The rule then controls how we can introduce a monadic resource in the set of hypotheses. Again reading it top-down, if we have a proof that from a set of resources that contain a certain resource A we can produce a monadic resource $\Diamond B$, then we can produce the same resource if we take the same set of resources and replace A with a monadic resource that encapsulates A.

Again, looking at the lambda term associated with the rule gives us an idea of how this replacement is performed. We assume that the original proof is encoded in the term

[12] To be precise, the unit may introduce only neutral information that acts as a multiplicative unit with respect to bind.

t associated with $\Diamond B$, and that the original resource A contributes a term to t that we represent with variable x. Now, if the lambda term associated with the monadic resource $\Diamond A$ is y, we can extract from it the encapsulated value that corresponds to A and give it the name x so that it is substituted in t for the original x. In terms of proof search (i.e. looking at the rule bottom-up), the left rule ensures that a monadic resource on the left which is not consumed directly by some negative context (such as those set up on the left-hand side of an implication) is matched by a monadic layer in the result formula. This is because we do not in general have a function that maps from the monadic layer to the non-monadic one.

Here are some useful theorems that exemplify some ways in which we are going to use monads:

- The first worked out proof of a theorem in our logic represents the combination of a resource that takes a certain argument, say A, consumes it and returns a combined resource B, something we represent in our system as $A \multimap B$, and a resource that is the required A but wrapped inside a monadic layer, which we represent as $\Diamond A$. We prove that from these two resources we can derive a B resource wrapped in a monad: $A \multimap B, \Diamond A \vdash \Diamond B$. The proof is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : A \vdash y : A}\, id \quad \dfrac{}{x : B \vdash x : B}\, id}{f : A \multimap B, y : A \vdash f(y) : B} \multimap L}{f : A \multimap B, y : A \vdash \eta(f(y)) : \Diamond B} \Diamond R}{f : A \multimap B, m : \Diamond A \vdash m \star \lambda y.\eta(f(y)) : \Diamond B} \Diamond L$$

  The lambda term associated with the right hand side of the sequent at the root of the tree gives us an idea of how the resources are combined: first the monadic argument, represented in the term by m, is evaluated according to the specific kind of monad we are dealing with, then the result is bound to the name y by the bind operator which also takes care of carrying over the monadic layer and whatever additional information/dependency was associated with m, and finally the value named y is passed to the argument consuming resource, f, and the result is wrapped in a neutral monadic layer by unit. Notice that this last step is necessary as in general we do not have a way to "exit" from a monad, also because the m resource may have introduced some additional semantic material that we do not want to lose.

- The second theorem we discuss in detail is a simplified version of many of the examples presented in the paper. Here we combine an argument-consuming resource that requires a monadic argument, as in the case of our characterization of verbs like *believe* or *love*, with a monadic argument.[13] Assuming that the argument is represented as $\Diamond A$ and the functional resource as $\Diamond A \multimap B$ we can show that there are two different proofs for the following theorem: $\Diamond A \multimap B, \Diamond A \vdash \Diamond B$.

  In the first proof the argument is passed directly to the functional resource and the final result is wrapped in a monadic layer by unit:

[13] If the argument is not of a monadic type we can always lift it "for free" using the unit.

$$\dfrac{\dfrac{\dfrac{}{m:\Diamond A \vdash m:\Diamond A}\,id \quad \dfrac{}{x:B \vdash x:B}\,id}{f:\Diamond A \multimap B, m:\Diamond A \vdash f(m):B}\multimap L}{f:\Diamond A \multimap B, m:\Diamond A \vdash \eta(f(m)):\Diamond B}\Diamond R$$

In the second proof the argument is first evaluated, the result is bound to the name y and then, after lifting it into a monad using unit, it is passed to the functional resource, and the result of their application is lifted in turn into a monad:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y:A \vdash y:A}\,id}{y:A \vdash \eta(y):\Diamond A}\Diamond R \quad \dfrac{}{x:B \vdash x:B}\,id}{f:\Diamond A \multimap B, y:A \vdash f(\eta(y)):B}\multimap L}{f:\Diamond A \multimap B, y:A \vdash \eta(f(\eta(y))):\Diamond B}\Diamond R}{m:\Diamond A, f:\Diamond A \multimap B \vdash m \star \lambda y.\eta(f(\eta(y))):\Diamond B}\Diamond L$$

Finally notice that the fact that we require generation of a monadic result from the combination of the two resources is crucial in obtaining two readings. If we try to generate a proof without the final monadic layer we obtain a single reading that is (roughly) equivalent to the first reading discussed above:

$$\dfrac{\dfrac{\dfrac{}{m:\Diamond A \vdash m:\Diamond A}\,id \quad \dfrac{}{x:B \vdash x:B}\,id}{f:\Diamond A \multimap B, m:\Diamond A \Rightarrow f(m):B}\multimap L}$$

- $\Diamond\Diamond A \vdash \Diamond A$, this basically says that no matter how many layers of monads we have we can always compress them to a single one (this is nothing but the definition of join in terms of bind!):

$$\dfrac{\dfrac{}{x:\Diamond A \vdash x:\Diamond A}\,id}{m:\Diamond\Diamond A \vdash m \star \lambda x.x:\Diamond A}\Diamond L$$

1.8 To a Categorial Grammar (ok this is becoming confusing)

At this point we could be done but let's go one step further and also provide a grammatical framework to play with monads. We are going to use a categorial grammar type of formalism, as it replicates quite directly the format of the logical calculus introduced in the previous section. What we have to do is basically define a structure for the left-hand side of our sequents and split the linear implication into two directional implications.

The structure we will assume is very simple, we will deal with strings of symbols (rather then trees), meaning that our grammar will have associativity baked in. And as far as the rule for implications, we present everything in figure 1.3 (notice that we write the argument of a directional implication always "under" the slash, meaning that $A\ B$ represents an expression looking for an A on the left to generate a B, and symmetrically B/A looks for an A on the right).

This framework has its limitations in terms of the language that it can parse but it should suffice for our purposes.

$$\frac{}{x : A \vdash x : A}\, id \qquad \frac{\Gamma \vdash B \qquad B, \Delta \vdash C}{\Gamma, \Delta \vdash C}\, Cut$$

$$\frac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda x.t : A \backslash B}\, \backslash R \qquad \frac{\Delta \vdash t : A \qquad \Gamma, x : B, \Theta \vdash u : C}{\Gamma, \Delta, y : A \backslash B, \Theta \vdash u[y(t)/x] : C}\, \backslash L$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : B / A}\, / R \qquad \frac{\Delta \vdash t : A \qquad \Gamma, x : B, \Theta \vdash u : C}{\Gamma, y : B / A, \Delta, \Theta \vdash u[y(t)/x] : C}\, / L$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u \rangle : A \otimes B} \otimes R \qquad \frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, \langle u, v \rangle : A \otimes B, \Delta \vdash t[x/u, y/v] : C} \otimes L$$

$$\frac{\Gamma \vdash x : A}{\Gamma \vdash \eta(x) : \Diamond A}\, \Diamond R \qquad \frac{\Gamma, x : A, \Delta \vdash t : \Diamond B}{\Gamma, y : \Diamond A, \Delta \vdash y \star \lambda x.t : \Diamond B}\, \Diamond L$$

Figure 1.3 Type-logical grammar with monadic modality as sequent calculus.

Lecture 2

Multidimensional Semantics

2.1 Pooping Dogs and Chicago Blues[1]

The heteregenous group of expressions that contribute to conversation by adding slightly off-topic information has recently received a lot of attention from the semantics community, especially since the publication of Potts [2005]. This is probably because, in the words of Barker et al. [2010], these expressions "challenge traditional conceptions of compositionality", but also because they seem to live on both sides of the semantics-pragmatics divide. In this lecture we are going to see how we can solve the challenges that these expressions pose using monads.

The reason why these expressions are problematic for standard approaches to compositionality is well illustrated by the following two examples:

(2.1) 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.

(2.2) 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

The sentence uttered by A in (2.1) seems to convey the information that the majority of the dogs in the neighbourhood defecate on the lawn of the speaker together with the fact that A has a generally negative attitude towards the neighbourhood dogs (or just the defecating ones) and/or their defecatory habits. The information about A's attitude is evidently conveyed by the two expressives *fucking* and *damn*. However the reply of B seems to target only the first piece of information. B seems to challenge only the fact that the neighbourhood dogs defecate on A's lawn, not the attitude of A towards the dogs.

Similarly in (2.2) the information conveyed by the appositive *the bluesman from Tennessee* cannot be negated by B by replying with *No, that's not true*, as that would target

[1] This lecture is an updated version of Giorgolo and Asudeh [2012b].

only the proposition that John Lee Hooker appeared in the *The Blues Brothers*. Instead B would have to resort to a different conversational strategy, first agreeing with A that John Lee Hooker appeared in the movie, but then adding that the information about his birth place is wrong.

On the basis of cases like those above and other similar considerations, Potts [2005, 2007] has proposed that expressions such as non-restrictive relative clauses, parentheticals, nominal appositives and expressives require the postulation of more than one level of semantic content. Potts calls these levels *dimensions* and identifies, for the class of expressions under discussion (at least), two distinct dimensions:

1. an 'at-issue' dimension, which represents the aspect of meaning that is under discussion and is sensitive to logical operators such as negation; in the examples above the content of the main clauses (that is excluding the content of the expressive and the nominal appositive) contributes to the at-issue dimension and is the target of the negative replies by B,

2. a 'side-issue' dimension, also known as the 'Conventional Implicature dimension', represents an aspect of meaning that contributes information that is speaker-oriented, often peripheral, and not under discussion or up for grabs; the expressives content in (2.1) conveys clearly speaker-oriented information (A's feelings towards the neighbourhood dogs) to which the interlocutor has limited access and therefore cannot easily discuss; in the case of the appositive in (2.2) the birth place of the musician is introduced as a kind of off-topic comment, not as part of the central discussion about his appearance in a movie.

However not all dimensions are born equal. In fact, a crucial aspect of Potts's analysis is that the interactions between the at-issue and the CI dimensions are restricted in terms of flow of information. According to the theory set up by Potts, semantic content can flow from the at-issue dimension to the CI dimension but not *vice versa*. In other words, the interpretation of expressions contributing to the CI dimension can reuse semantic material introduced in the at-issue dimension but there are no lexical items that recycle material belonging to the CI dimension and introduce it in the at-issue one.

This last claim has been recently called into question by AnderBois et al. [2010, 2015] in light of examples like the following:

(2.3) John$_1$, who by the way almost destroyed his$_1$ car yesterday, has bought a motorcycle, too.

What we observe is a complex network of interactions between the two dimensions. The entity introduced by *John* and the associated discourse referent are respectively used as the anchor point for the relative pronoun heading the relative clause and as the antecedent for the anaphoric pronoun *his* as indicated by the indexes. This type of interaction is expected in Potts's theory. However, we also have unexpected information flow from the side-issue content to the at-issue content, since the presupposition triggered by *too* is satisfied by the information contributed in the side-issue appositive, that John has another vehicle. This seems to contradict the restriction that Potts imposes on the

direction of the flow of information, if we assume a loose, but reasonable, interpretation of what 'flow of information' is and take it to include the satisfaction of a presupposition.

In this lecture we argue for a treatment of conventional implicature in terms of multiple dimensions, in agreement with Potts [2005, 2007] and *contra* AnderBois et al. [2010, 2015]. To explain the data discussed by AnderBois et al. [2010, 2015] we will propose a two stages analysis: during the compositional process the two dimensions are kept separate and obey Potts [2005]'s restrictions regarding the flow of information, while in the post-compositional phase (i.e. when anaphoric relations are established and discourse consistency checks are performed) the interactions between the dimensions are free.

The main challenge is to create a compositional model of this procedure and the main contribution of this paper is the introduction of such a model. We will see how monads can help us do exactly that. The specific monad we will use is the one constructed around the functor that maps each type τ to the type of pairs of elements of τ and elements of the monoid formed by a set of propositions. Just to recapitulate, remember that a monoid is formed by a set M closed under an associative binary operation $\cdot$ and that has an element ϵ such that for all elements x of M we have $x \cdot \epsilon = \epsilon \cdot x = x$. In our case we can choose among different monoids, that are more or less equivalent for our purposes: we can either take the set of all lists/sets of propositions together with list concatenation / set union and the empty list / empty set as the identity element, or, if we are only interested in truth values, the set $\{\top, \bot\}$ with $\wedge$ and $\top$ as respectively the binary operation and the identity element.

A more general proposal is the one of Barker et al. [2010]. Their model has many points of contact with the one presented here and shares very similar goals. Barker et al. [2010] use a continuation-based approach to model the restricted interaction between dimensions. While continuations are provably equivalent to the general monadic framework we use for our model, we will argue that our approach is preferable in that it zeroes in on the essential properties of conventional implicature and predicts the full range of interactions without having to depend on the full power of monads.

The lecture is structured as follows. In section 2.2 we review the arguments in favor and against a multidimensional semantics for conventional implicature and show why a multidimensional semantics is indeed necessary. In section 2.3 we review two previous proposals to model the interaction between dimensions in a compositional fashion. In section 2.4 we discuss how plan to use monad to model conversational implicatures and exemplify their use in section 2.5 with a fully worked out semantic derivation. We conclude in section 4.5 with some closing remarks.

2.2 Interdimensional Meaning Interaction

We have already discussed that AnderBois et al. [2010, 2015] reject the Potts's multidimensional analysis on the basis of a number of circumstances, initially discussed by Potts [2005, 52ff.], in which at-issue content seems to require access to side-issue content. We repeat here some of the cases they analyze:

1. **Presupposition**

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

This example is comparable to the one previously discussed. Here the presupposition that Mary has some additional musical talent besides being a good singer is supported by the information conveyed by the nominal appositive *a good drummer*

2. **Anaphora**

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

   In (2.5) the pronoun *her* founds its antecedent in the non-restrictive relative clause, in a way leaking information from the CI dimension to the at-issue dimension.

3. **VP ellipsis**

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

   Similarly here the elided VP is first introduced in the relative clause, i.e. in the CI dimension.

4. **Nominal ellipsis/anaphora**

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

   Also here the nominal ellipsis (or the pronominal use of *six*) seems to break the Pottsian rule of information flowing only from the at-issue to the CI dimension.

According to AnderBois et al. [2010, 2015] this kind of data makes the analysis of Potts untenable. The reasoning is that given that the multidimensional treatment of conventional implicatures is founded on the intuition that the dimensions are fundamentally independent (or at least the at-issue dimension is independent of the side-issue one) there is no way to reconcile it with the type of data just discussed. Their conclusion is therefore that there is only *one dimension* of meaning and that conventional implicatures live happily together with at-issue meaning in this dimension.

Clearly a purely one-dimensional approach is faced with major challenges when trying to explain the other kind of data, exemplified in the introduction by the two mini-dialogues (2.1) and (2.2). In a unidimensional approach there is in fact nothing preventing side-issue content from interacting with logical operators such as negation. The solution that AnderBois et al. [2010, 2015] propose is that, instead of two dimension of meaning, there are *two modes* of discourse update, one for at-issue material and one for side-issue material. At-issue material is *proposed* and open for correction, questioning, etc. Side-issue material is instead *imposed* and the update eliminates possible interpretations that are inconsistent with the side-issue meaning. In this way, the data in (2.4–2.7) is explained on the basis of a full interaction between dimensions, and at the same time the fact that in (2.1) and (2.2) the replies of B do not target the side-issue comments is accounted for on the ground of its imposed status.

Our understanding is that this analysis is however not capable of accounting for a number of other circumstances. The first problem we encounter when trying to apply

AnderBois et al. [2010, 2015] is when conventional implicatures interact with logical operators that are not at the discourse level. AnderBois et al. [2010, 2015] do not explain how discourse updates are generated from syntactic structures but given their assumption of unidimensionality we assume that a single logical form is derived. In light of this hypothesis consider the following example:

(2.8) Luke Skywalker is so gullible that he believes that Jabba the Hutt, a notorious scammer, is a trustworthy business partner.

Here clearly the information that Jabba is a scammer is not part of Skywalker's beliefs, therefore projecting somehow outside of the scope of *believes*. To obtain such an effect with a single logical form we would have to postulate an *ad hoc* rule that moves the content of the appositive outside of any logical operator above it in the derivation. Similar rules would be needed for all other expressions conveying conventional implicatures leading to an undesirable proliferation of scope-escaping devices. Alternatively the appositive could be somehow marked in the logical form as requiring special treatment during discourse update, but then it would be difficult to see how such an approach would be relevantly different from Potts's analysis.

A unidimensional approach also seems incapable of correctly predicting the patterns of interaction between the at-issue and side-issue meaning. The interactions are in fact not completely free, but instead seem to respect limitations that we observe at the discourse level. This is also true for the cases in which information flows from the at-issue to the side-issue dimensions/modes. Consider the following examples:

(2.9) All Cairo taxi drivers$_1$, who by the way painted their$_1$ taxis red in protest, are on strike.

(2.10) *Every Cairo taxi driver$_1$, who by the way would threaten me with his$_1$ gun, is on strike.

If we assume a single logical form such that both relative clauses are in the scope of the respective quantifiers, then it is not quite clear why the possessive pronoun *their* in (2.9) can be bound by the quantifier *all*, while the same is not true in (2.10). What we observe here seems to be more in line with an analysis in which the two pronouns are considered anaphoric (i.e. as part of two distinct sentences), so that their acceptability is dependent on their ability to establish an anaphoric relation with their antecedents. In the case of (2.9) *their* manages to find its antecedent (the totality of Cairo taxi drivers) thanks to the well known ability of plurals to establish discourse referents [Nouwen, 2003]. In (2.10) instead the pronoun *his* cannot be anaphorically related to referent introduced by the universal quantifier as it is local to its scope.

We propose a different solution based on the observation that the interaction between at-issue and side-issue content is limited to a certain class of discourse related phenomena and follows the same patterns we observe when dealing with proper discourse fragments. We start from the intuition that at the level of logical form generation we have two distinct semantic dimensions. The interaction between these two dimensions follow Potts's principle of limited interaction: at-issue meaning resources can be re-used in the CI

dimension but side-issue content never leaks into the at-issue dimension. Once the logical form is complete (and possibly contains things like free variables and presuppositions to be satisfied) the boundaries between dimensions are lifted and the resolution of discourse-related uncertainties can take place, under the condition that we have two propositions that function at all effects as two distinct discourse segments.

This analysis correctly accounts for the data. The non-interaction between conventional implicatures and logical operators is explained, following Potts [2005], in terms of multidimensionality. For instance if we apply our analysis to example (2.8) we obtain two distinct propositions, one expressing the fact that Skywalker is so gullible that he believes that Jabba is a good business partner and the other expressing the fact that Jabba is a notorious scammer. The limited interactions we observe in examples (2.4–2.7) are instead explained as a discourse-level phenomena. This also allows us to predict the acceptability of (2.9) and the non-acceptability of (2.10).

The next step is to explain how we can build in a compositional fashion the two meaning components while keeping the flow of information under control. Before describing our approach we will review two previous proposals in further detail.

2.3 Conventional Implicature and Compositionality

The analysis by Barker et al. [2010] builds on the idea of continuation-based semantics, a promising approach explored by the authors in a number of settings. The use of continuations allows them to model with a single device the full stack of operations needed to account for the data under discussion. The analysis is quite complex and we report here only the main characteristics of it.

The general idea is that the interpretation of a sentence is a function not only of its component parts but also of the speaker uttering it (for instance necessary to obtain the correct attribution of judgements in case of expressives) and the current common ground, i.e. the collection of propositions introduced in the discourse up to the expression under consideration. The output of the interpretation of a sentence is a pair of propositions that correspond to the at-issue and side-issue contributions to the common ground. Therefore the *return type* of their semantic objects is a function from a speaker to a function from an input common ground to a pair of propositions.

The lexical meanings are lifted to a continuation-passing form, where each semantic object is wrapped into a function that takes its context as an argument and applies the original meaning to it as its argument. The meaning of at-issue expressions is however in a way different from that of expressions conveying conventional implicatures. In fact, in the first case the return type of the context is opaque to the lexical entry; that is, the lexical entry does not contain the information that it is a function from a speaker to a function from a common ground to a pair of propositions. The return type is instead part of the lexical item that contributes to the CI dimension. In this way, these lexical item can operate on it and, for example, contribute to the proposition corresponding to the CI dimension. Therefore in Barker et al. [2010]'s approach the flow of information is completely controlled at the level of types.

Another approach to modeling the composition of at-issue and side-issue meaning is

the one of Arnold and Sadler [2010, 2011]. Arnold and Sadler [2011]'s analysis is cast in the framework of LFG and follows a suggestion by Potts [2005, 85ff.] in capturing multidimensionality directly in the logic for composition. In the context of the glue logic of LFG's Glue Semantics, where the glue logic is a non-commutative fragment of linear logic, this means that, not dissimilarly to what Barker et al. [2010] do, the glue logic terms on the right side of glue meaning constructors are used to distinguish lexical resources that contribute different types of semantic content. For readers not familiar with Glue Semantics, this means that the job controlling the flow of information is deferred to the types and the logic determining their composition. In contrast we will see that the monadic approach leaves the type signature of our lexical items (superficially) unchanged and moves the responsibility of keeping track of the information flow to the meaning terms. The proposal by Arnold and Sadler [2010] also presents a number of downsides mainly related to the limited expressive power of the logical terms of Glue Semantics. In particular the problems are all related to the fact that to implement the coupling of at-issue and CI material we have to resort to a *tensor product* connective that presents a number of problems:

1. In principle, it might be necessary to propose more than two dimensions. In such a case, the commutative tensor conjunction in linear logic does not provide enough structure to properly distinguish between dimensions or to refer to information in a particular dimension subsequently.

2. The lack of structure in the tensor conjunction makes it difficult to control at-issue/side-issue interactions of the kind discussed above.

3. Tensors in proof goals make it more difficult to state the correct condition on proof termination and therefore potentially lose some of the linguistic leverage provided by linear logic's resource sensitivity [Asudeh, 2004, 2012].

2.4 Monads for Conventional Implicature

In this section we introduce the technical machinery implementing our analysis. The main ingredients of our implementation 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, while the second component is not a proposition but rather a *collection of propositions*, containing all the side-issue information conveyed by the sub-parts of 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.

As anticipated, the monad we are going to 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:

$$\Diamond(f) = \lambda\langle x, Q\rangle.\langle f(x), Q\rangle \tag{2.11}$$

The monoid functions as a storage for side-issue propositions, accumulating them in a set. The identity $\emptyset$ correspond 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:

$$\eta_A(x) = \langle x, \emptyset\rangle \tag{2.12}$$

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

$$\begin{array}{ccc} Id(A) & \xrightarrow{\eta_A} & \Diamond A \\ {\scriptstyle Id(f)}\downarrow & & \downarrow{\scriptstyle \Diamond(f)} \\ Id(B) & \xrightarrow[\eta_B]{} & \Diamond B \end{array}$$

and here is the proof that this is the case:

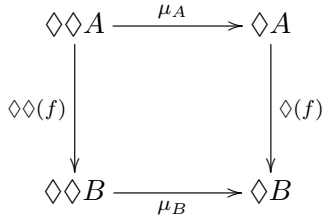
$$\eta_B(f(x)) = \langle f(x), \emptyset\rangle = \Diamond(f)(\langle x, \emptyset\rangle) = \Diamond(f)(\eta_A(x))$$

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:

$$\mu_A(\langle\langle x, P\rangle, Q\rangle) = \langle x, P \cup Q\rangle \tag{2.13}$$

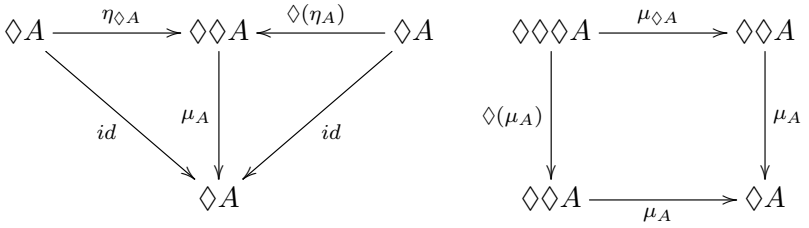
Let's check again that this is indeed a natural transformation:



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:



(2.14)

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

$$m \star f = \mu(\Diamond(f)(m)) \tag{2.15}$$

In our case we obtain the following definition:

$$\langle x, Q\rangle \star f \rightsquigarrow \mu(\langle f(x), Q\rangle) \rightsquigarrow \mu(\langle\langle y, P\rangle, Q\rangle \rightsquigarrow \langle y, P \cup Q\rangle \tag{2.16}$$

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 $f(x)$:

$$\langle x, P\rangle \star f = \langle \pi_1(f(x)), P \cup \pi_2(f(x))\rangle \tag{2.17}$$

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

$$m \star \eta = m \tag{2.18}$$

$$\eta(x) \star f = f(x) \tag{2.19}$$

$$(m \star f) \star g = m \star (\lambda x.f(x) \star g) \tag{2.20}$$

The first is easy:

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

Here is the proof for the second rule:

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

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

$$m \star f = \langle \pi_1(f(\pi_1(m))), \pi_2(m) \cup \pi_2(f(\pi_1(m)))\rangle \tag{2.21}$$

the only difference being that here we decompose the first argument using projection functions instead of our sloppy pattern matching. First let's see what the left hand side of the rule reduces to:

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

And here is the right hand side reduction:

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

As you can (probably with some difficulty) see the two reductions end in the same term as expected.

2.5 Analysis

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

(2.22) 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).

Lexicon

The mini lexicon we use is show in table 2.1. 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 play so much 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 simply as an identity function that at the level of syntax simply changes the type to the preferred one (the one of a non-restrictive relative clause).[2] 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 of the lexicon is the denotation of *COMMA*. Its denotation it's 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. 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:

$$\mathtt{p} = \langle 1, \{p\} \rangle \tag{2.23}$$

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 2.1 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 refer you

[2] 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*.

to the online tool. Instead let's look at 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:

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

Suggested exercises:

- Prove that the diagrams in 2.14 commute.
- Extend the lexicon with expressives like *damn* and *fucking* and test it with the online tool.
- Extend the lexicon to deal with non-restrictive relative clauses (e.g. *Ash, who was born in Ottawa, is a big Liverpool fan*) and test your entries with the online tool. (By the way, Ash is a Liverpool fan, but he wasn't born in Ottawa)

Lecture 3

Perspectives

3.1 Introduction[1]

An important problem in the philosophy of language and the linguistic study of meaning concerns co-referential terms and substitutability in different contexts. In the modern context, this problem is commonly associated with Frege [1892], and is often called *Frege's puzzle*. The puzzle can be presented in various ways, but its essence can be captured as follows: given two co-referential linguistic expressions, why is it that in certain linguistic contexts substitution of one expression for the other is truth-preserving, while in others it is not?

For example, given that (3.1) is true, since Hesperus and Phosphorus are different names for the planet Venus, how is it that (3.2) can be true while (3.3) is false?

(3.1) Hesperus is Phosphorus.

(3.2) Kim believes that Hesperus is a planet.

(3.3) Kim believes that Phosphorus is a planet.

Alternatively, we could characterize the puzzle by observing that a sentence like the following can be true without entailing that Kim does not believe a tautology:

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

Frege's own solution was that in addition to a reference, nominals have a sense, or 'mode of presentation', and that in certain contexts, such as those involving propositional attitudes, it is these distinct senses that block substitutability. Frege's puzzle is thus clearly related to the problem of 'referential opacity' in the study of propositional attitudes [Quine, 1953, 1956, 1960]. Fregean senses are not the only way to construe modes of presentation [e.g., Schiffer, 1990, Fiengo and May, 1998] and the notion that names, in particular, have a mode of presentation is not universally accepted. For example, Kripke [1972, 1979, 1980] mounts an influential defence of a Millian theory of names as simply referential.

[1] These notes are a lightly edited/redacted version of Asudeh and Giorgolo [2015].

We focus on linguistic aspects of substitutability/opacity. We take it for granted that there is an empirical phenomenon to be explained here — differing truth value judgements despite substitution of co-referential terms — and offer a formal mechanism for capturing and explaining it. We follow Saul [1997, 2007] in observing that problems of substitutability also arise in 'simple sentences'. Our analysis captures these cases, too. Moreover, we also focus on cases of differential interpretation of the *same* expression [Kripke, 1979, Castañeda, 1989, Fiengo and May, 1998]. Lastly, we briefly indicate how our analysis could give insight into cases other than referential expressions, as discussed by Kripke [1979]. The monadic mechanism we propose provides the beginnings of a general formal semantics of what we might informally call *perspective*.[2]

3.2 The Scope of the Problem

The substitutability puzzle is standardly characterized as involving two factors: 1. embedding under a modal or propositional attitude expression, such as *believe*; and 2. co-referential but distinct terms, such as *Hesperus* and *Phosphorus*. This is just how we presented things earlier. However, it has been shown in the literature that neither of these factors is necessary for the substitutability puzzle or related puzzles to arise.

3.2.1 Simple Sentences

With respect to the first factor, Saul [1997] points out that the lack of substitutability can hold even in 'simple sentences' that 'contain no attitude, modal or quotational constructions' [Saul, 1997, 102, fn.1]. Assuming it is common knowledge that Clark Kent is Superman's secret identity, she notes that if (3.5) is true, substitution of *Clark Kent* for *Superman* seems to render (3.6) false [Saul, 1997, 102, (1) & (1*)]:

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

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

With respect to this pair, an obvious issue presents itself: Are these sentences actually semantically non-distinct, with one just being a better way to package the information (in some sense that would need to be made more precise)? That is, it seems that we could say these sentences are in fact truth-conditionally equivalent (so (3.6) is not false when (3.5) is true). This is in fact what Braun and Saul [2002] and Saul [2007] argue: namely, what is mistaken is our intuition that (3.5) is true while (3.6) is false. Braun and Saul [2002] and subsequently Saul [2007, 124ff.] offer an account of this mistaken intuition based on a psychological account of how we store and access names (and other referential expressions).

[2]This notion of perspective may well be related to notions of perspective or point of view in other phenomena, such as demonstratives and indexicals (e.g., Kaplan, 1989, Schlenker, 2003), *de se* attitudes (e.g., Lewis, 1979, Chierchia, 1989, Pearson, 2013), logophoricity and related pronominal interpretations (e.g., Sells, 1987, Hagège, 1974, Kuno, 1987, Oshima, 2006, Sundaresan, 2012), illocutionary adverbs (e.g., Austin, 1975, Krifka, 2001, 2014, Ernst, 2009), expressivity (e.g., Potts, 2005, 2007, Gutzmann, 2012, Gutzmann and Gärtner, 2013), and predicates of personal taste (e.g., Lasersohn, 2005, Stephenson, 2007a,b, MacFarlane, 2014).

The following sentences display a contrast, whether or not both the speaker and her audience are enlightened that Peter Parker is Spider-Man:[3]

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

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

Even given an enlightened speaker and audience, (3.7) is contradictory: killing Spider-Man entails killing Peter Parker (assuming it is indeed Peter Parker who is Spider-Man at the time; i.e. there has been no passing of the mantle or any such thing). Of course, for the unenlightened the sentence would not be perceived as contradictory: it's consistent for Dr. Octopus to kill two different people. Moreover, there is no such contrast with (3.8): *murder* necessarily involves intention and the sentence is not contradictory even for the enlightened. It is not clear to us how the Saulian could capture these facts, which clearly rest on a distinction between *kill* and *murder*.

The following sentence is also potentially problematic for Saul's view:

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

Let us assume that the time of evaluation is a point in the stories before Mary Jane knows that Peter Parker is Spider-Man. According to the theory presented in Saul [2007], this sentence is simply false, but that seems to entirely set aside Mary Jane's say in the matter, which strikes us as problematic. It would be very strange to insist that if Mary Jane loves Peter Parker, then she really does love Spider-Man: she certainly wouldn't agree to that. Rather, sentence (3.9) crucially involves Mary Jane's perspective or point of view.[4]

We should forestall potential attempts to reduce our theory of perspectives to a theory of *guises* [Castañeda, 1972, 1989, Heim, 1998], despite superficial similarities to and a shared empirical base with Castañeda's theory. Our approach is distinct from one where a sentence like (3.9) is interpreted as simply saying that Mary Jane loves only one *guise* of the entity that corresponds to Peter Parker but not another one. First, one might object that people love entities not guises. Something like this seems at play in the criticism of MacColl [1905a,b] by Russell [1905, 491], although we suspect that a Neo-Meinongian such as Castañeda would not have found this ultimately convincing. As linguists, we feel neither prepared nor compelled to enter this debate, but a simple guise-based theory in fact also makes false empirical predictions.

If it is indeed the case that different co-referring expressions simply pick out different guises of the same individual, then a sentence like (3.7) should have a non-contradictory reading, but this does not seem to be the case (again assuming it is indeed Peter Parker who is Spider-Man at the time). Killing Spider-Man simply seems to entail killing his

[3]We prefer to use Spider-Man in our examples, because Superman is frankly kind of boring, but also because the Peter Parker/Spider-Man case involves a different (yet still familiar) set-up: it is not as clear which is whose secret identity, since Peter Parker is as much the "main character" in those stories as Spider-Man is. This avoids the problem of Pitt's [2001] concept of *primum egos*, as discussed by Saul [2007, 31–34,140]. It also avoids the problem that both Superman and Clark Kent are in fact secret identities of a third identity, Kal-El, a Kryptonian refugee [Saul, 2007, 31–34].

[4]This seems at least superficially similar to Castañeda's story of Greta Bergman and Oscar A. A. Hecdnett, which crucially involves Hecdnett's point of view [Castañeda, 1989, 21–30].

alter ego, Peter Parker. And once again, the theory must capture the difference between, for example, *kill* and *murder*, given the contrast in (3.7) versus (3.8). Although we characterize *murder* as involving 'intention', it is not a propositional attitude verb and there is no obvious evidence of embedding. We discuss this further in section 3.5.

Lastly, whatever analysis we give must not lose sight of the fact that there are incontrovertible cases of contradiction that must still be derivable, such as the following:

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

(3.11) #Bill shares a birthday with Spider-Man, but he doesn't share a birthday with Peter Parker.

What unites *murder* and *love* versus *kill*, *punch*, and *share a birthday with* is the fact that, for the former pair, the subject/agent's perspective is part of the interpretation.

3.2.2 Non-Distinct Terms but Distinct Beliefs

Kripke [1979] presents a puzzle that is closely related to the substitutability puzzle, but which relates to the second factor mentioned above: whether the terms involved must be distinct. He considers the case of 'phonetically identical tokens of a single name'. He provides the example of an individual, Peter, who has learned that *Paderewski* was the name of an accomplished Polish pianist. The following then seems true:

(3.12) Peter believes that Paderewski had musical talent.

Peter then hears of a Polish politician named *Paderewski*, and concludes that this is a different person, since he has no reason to believe that politicians make good musicians. Given that in fact the same Paderewski was in fact both a politician and a pianist, is the following true or not?

(3.13) Peter believes that Paderewski had no musical talent.

Kripke [1979] argues that this is a true paradox and we can neither conclude that (3.13) is true nor false, given the situation.

Fiengo and May [1998] deny this conclusion on the basis of a theory of reference that crucially holds that names do not directly refer but only do so once part of linguistic expressions, which bear distinguishing indices, such as '$[_{\mathrm{NP}_1}$ Paderewski]' and that what the speaker believes is characterized by statements of the following form [Fiengo and May, 1998, 388]:

(3.14) $\ulcorner$'$[_{\mathrm{NP}_i}\ X]$' has the value $\mathrm{NP}_i\urcorner$

They also propose the following principle:

(3.15) **Singularity Principle**
If cospelled expressions are covalued, they are coindexed.

For Fiengo and May, then, there are two distinct Paderewski indexations at play for Peter, which means that the two "cospelled" instances of Paderewski are not covalued, given the Singularity Principle. We will return to Fiengo and May later in the lecture.

The informal theory that Fiengo and May [1998] put forward is closely related to the Interpreted Logical Form theory of Larson and Ludlow [1993], who provide the following cute and memorable, but ultimately unconvincing example [Larson and Ludlow, 1993, 336]:

> **Context:** Jason is from New York and does not know how the name *Harvard* is pronounced in a Boston accent.

(3.16) Jason believes [Harvard is a fine school].

Using [harvard] to indicate Jason's pronunciation of *Harvard* and [hahvahd] to indicate the Boston pronunciation, Larson and Ludlow point out that, given this context, (3.17) is true, while (3.18), is false:

(3.17) Jason believes that [[harvard] is a fine school].

(3.18) Jason believes that [[hahvahd] is a fine school].

Why do we find this unconvincing? It seems to us that, for Jason, [harvard] and [hahvahd] are just different words. The fact that they are different pronunciations of the same word is etymological knowledge that is irrelevant to Jason's synchronic knowledge of language. Coincidence of spelling is similarly irrelevant — a criticism that applies to Fiengo and May's Singularity Principle, too (cf. 'cospelled expressions' in (3.15) above). Kripke in fact characterized things much more aptly when he wrote of 'phonetically identical tokens of a single name': homo*phony* is what's at stake, not homo*graphy*.

In our opinion, a more satisfactory analysis of these kinds of linguistic puzzles rests on disentangling two different phenomena that seem at play in Paderewski puzzles. Kripke's [1979] conclusion that we are dealing with a paradox seems to us motivated by the interplay between the perspectival dimension introduced by the verb *believe* together with the *ambiguous* nature of the name *Paderewski* in the context of Peter's lexicon. In this case we not only have different perspectives regarding the interpretation of a term (the speaker's and Peter's), but the two interpretations also have different *cardinalities*. Given that Peter can use the name Paderewski to refer to two different (from Peter's perspective) entities, in an example like (3.13) it is not possible to resolve whether we are talking about Peter's belief with regard to the pianist entity or the politician one. Therefore (3.13) seems to lack a determinate truth value: it is true with respect to Paderewski the politician, but false with respect to Paderewski the musician. We have competing interpretations, but each one is fully interpretable and can be assigned a truth value. Of course, this move itself only makes sense if the two instances of the name *Paderewski* in fact do not refer to one and the same entity for Peter, which is not possible for a Millian like Kripke, certainly on a Naive Millian view [Salmon, 1986], but is possible along the lines of our analysis in section 3.4. Furthermore, such an analysis easily generalizes to

interpretations of any cardinality: it would be no more problematic if Peter thought there were three Paderewskis, or four, or more.[5]

3.2.3 Identity Statements: Delusions and Mathematical Truths

The observations that homophonous terms and simple sentences can likewise lead to the substitutability puzzle and related puzzles is thus established in the literature. But it seems to us that we can drive the point home in an even simpler way, by starting with basic identity statements involving two homophonous tokens of the same name, avoiding accents and bypassing Paderewskis.

Statements such as the following are normally taken to be uninformative tautologies:

(3.19) Sandy is Sandy.

If this is true, then a statement like the following should mean that Kim does not believe a tautology:

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

Let us call the reading where Kim does not believe a tautology an *unsatisfiable* reading.

However, sentences like (3.20) also have satisfiable readings in the right context:

(3.21) **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.'[6]

In this context, it is clear that one instance of Sandy is interpreted from the speaker's perspective, call this $\mathbf{Sandy}_\sigma$ (where σ is the speaker index) and the other from Kim's, call this $\mathbf{Impostor}_{\mathtt{kim}}$. The speaker is then asserting that Kim does not believe that $\mathbf{Impostor}_{\mathtt{kim}} = \mathbf{Sandy}_\sigma$. In a sense, then, this is a linguistically simple, limiting case of the puzzles we have been looking at.

This type of expressions are not restricted to pathological cases. We can also construct similar examples in the case of mathematical terms, a domain that we would not expect to be open to interpretation. Consider the following piece of American history:

(3.22) **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).[7]

At the appropriate historical juncture, it is clear that the following sentence had a satisfiable reading:

[5] The case of the protagonist Ralph in Quine's [1956] well-known Ortcutt story is parallel.

[6] Wikipedia: `http://en.wikipedia.org/wiki/Capgras_delusion`

[7] Wikipedia: `http://en.wikipedia.org/wiki/Indiana_Pi_Bill`

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

Dr. Goodwin was clearly mathematically benighted, but given the context, it seems that (3.23) accurately reported his beliefs.

It may be tempting to explain these facts in terms of a *de re/de dicto* distinction based on compositional scope, as in Montague Semantics [Montague, 1973]. For standard examples, like the well known Hesperus and Phosphorus case, a number of convincing analyses based on a *de re/de dicto* ambiguity have been proposed in the literature. For instance, Aloni [2005] presents an appealing treatment of such cases in the context of an epistemic predicate logic [Hintikka, 1975]. However if we try to apply the same type of analysis to the cases discussed here, we encounter some problems.

Along these lines, we could try to analyze an example like (3.23) such that one instance of the name π gets a *de re* interpretation (the real ratio of a circle's circumference to its diameter), and the other a *de dicto* one (the rational number that Dr. Goodwin calls 'π'). The first problem we encounter is that, if we assume that proper names get translated to constants in the epistemic predicate logic, then both instances are interpreted in the scope of the modal operator, and therefore get a *de dicto* interpretation.

A standard move to solve this problem would be to assume that proper names denote scopal operators [Montague, 1973], such as quantifiers, built around their referents. This allows us to introduce two quantifiers that can scope either above or below the epistemic modality. We could therefore capture the meaning of (3.23) with the formula in (3.24), where $\Box_G$ is a doxastic belief operator relativized to Dr. Goodwin [Aloni, 2005].

$$\exists x.\neg\Box_G\exists y.x = 3.1415926535... \wedge y = 3.2 \wedge x = y \tag{3.24}$$

However, from a compositional point of view, arriving at this representation is not an easy task.

The formula in (3.24) assumes that the same name π gets instantiated with two different values (the irrational one and the rational one), but what would the lexical representation of this difference be? We could assume that π is ambiguous between these two meanings and that the expected reading emerges when each of the two meanings is picked. However, consider what would happen if Dr. Goodwin had had a rival — let's call him Dr. Badwin — in the Indiana General Assembly trying to push an alternative bill proposing that π equals 3.15. In that case, on the mooted approach we would be forced to also include 3.15 among the possible referents of π. But this would generate readings like (3.25):

$$\exists x.\neg\Box_G\exists y.x = 3.1415926535... \wedge y = 3.15 \wedge x = y \tag{3.25}$$

This formula would be true given the circumstances, because it is in fact not the case that, in all doxastically accessible belief worlds of Dr. Goodwin's, $3.1415926535... = 3.15$. But this is too weak an interpretation for Dr. Goodwin's actual beliefs: none of his belief worlds are such that he believes π is 3.15 — that's Dr. Badwin's belief. In other words, there is a stronger requirement on compositional interpretation than we would get, in the general case, by simply treating terms as ambiguous, tout court. Rather, they are potentially ambiguous in different ways for different speakers. One way to capture this is our method, explained below, of allowing interpretation to be anchored to different agents' potentially differing perspectives.

It is important to note that, like standard *de re/de dicto* approaches, our approach is also based on scope, but on the scope of operators in an enriched meaning language that represents perspectival semantics, not on the scope of quantifiers relative to attitudinal operators in some logical form or other representation of the syntax–semantics interface. This is what allows our system to deal with substitution in simple sentences, for which postulation of relevant attitudinal operators is not motivated [Saul, 1997, 2007].

3.2.4 Summary: The Space of Explananda

Intuitively, what the Capgras and Indiana Pi Bill cases share is a mix of the speaker's perspective with some other perspective: that of the subject of the sentence. Thus, it seems to us that the key to these puzzles, as mentioned above, is a notion of *perspective*, which can also potentially explain the lack of substitutability in simple sentences involving verbs like *love* and *murder*, as well as the standard cases of non-substitutability of co-referential terms in embedded contexts. If we cross the factors of same/distinct terms with simple/embedded context, we obtain the space of explananda in Table 3.1, with cells filled by examples from the previous sections.

	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 3.1 The space of explananda.

3.3 Formalization

In section 3.3.2 we present our proposed formalization. But before proceeding with our proposal we present, in section 3.3.1, an alternative formalization done in the style of the Logical Form semantics of Heim and Kratzer [1998]. This will provide a baseline against which we argue that our solution is preferable.[8]

3.3.1 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

[8]We would like to thank one of our anonymous reviewers for suggesting this alternative formalization.

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 (3.26); the perspective index for the interpretation of the composed expression is the same as that of its subexpressions.

(3.26) *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 (3.27) we present a revised version of the rule as discussed in Heim and Kratzer [1998, 186].

(3.27) *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:

$$[\![\text{Mary Jane}]\!]^{g,i} = \mathbf{mj}_\sigma \tag{3.28}$$

$$[\![\text{Peter Parker}]\!]^{g,i} = \mathbf{pp}_\sigma \tag{3.29}$$

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:[9]

$$[\![\text{Spider-Man}]\!]^{g,i} = \begin{cases} \mathbf{sm}_{\mathtt{mj}} \text{ if } i = \mathtt{mj} \\ \mathbf{pp}_\sigma \text{ if } i = \sigma \end{cases} \tag{3.30}$$

[9]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*.

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.[10] 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, which we discuss in more detail in section 3.4):

$$[\![\text{loves DP}]\!]^{g,i} = \lambda s.\mathbf{love}(s, [\![\text{DP}]\!]^{g,\kappa(s)}) \tag{3.31}$$

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:

$$[\![\text{punch}]\!]^{g,i} = \lambda o.\lambda s.\mathbf{punch}(s, o) \tag{3.32}$$

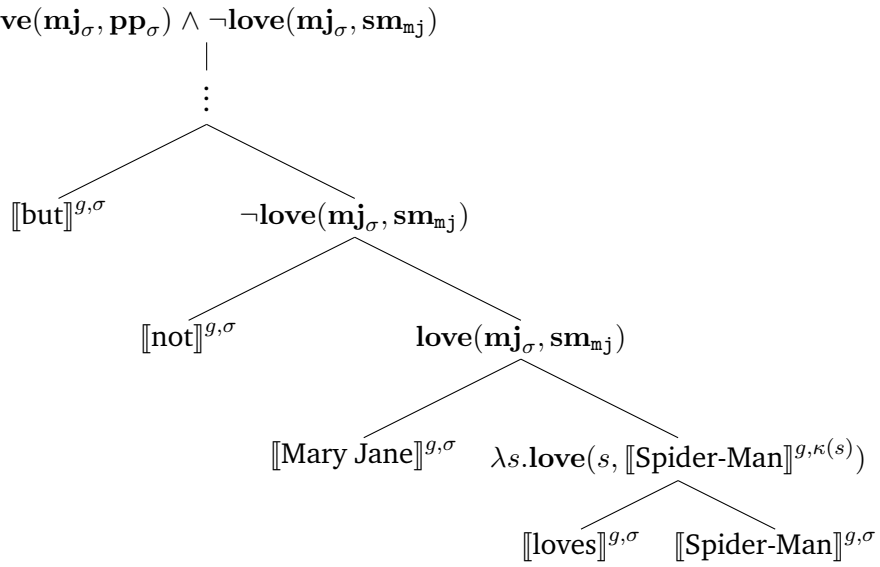
Equipped with this mini lexicon, we can sketch a preliminary analysis of an example like (3.9), repeated here as (3.33)

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

Our analysis is centered around the fact that (3.33) 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. 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:

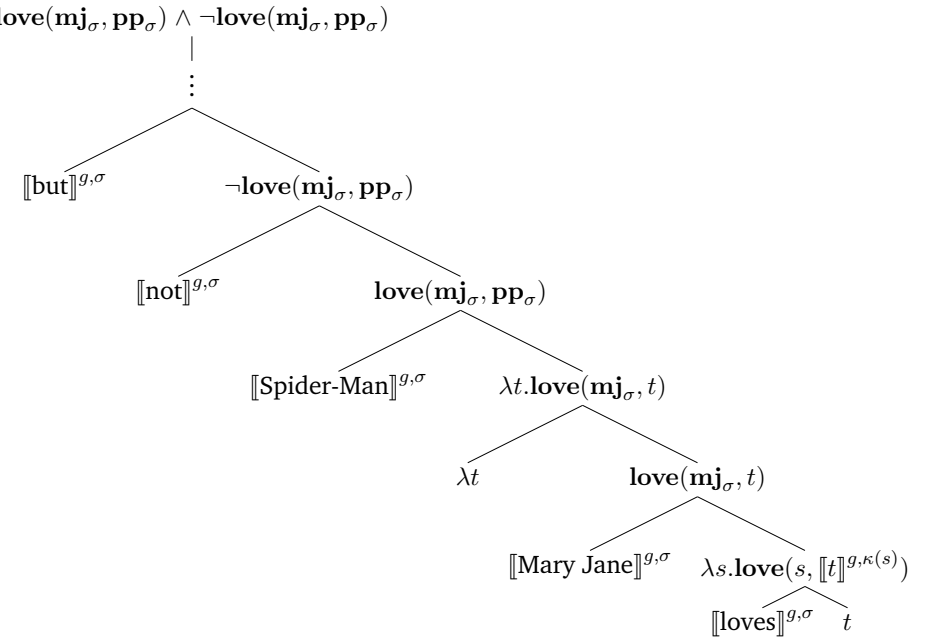
[10] In principle we could add the indices to the target meaning language, and this is indeed the choice we make in our alternative monadic implementation below. However in the case of standard Heim and Kratzer-style semantics we would still need to modify the rules for functional application and predicate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

(3.34)



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:

(3.35)



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 in the next section is preferable to the Logical Form semantics that we have just sketched. First of all, 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. 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*. 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. 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.

3.3.2 Formalization with Monads

Our actual formal proposal is a conservative extension of the simply-typed lambda calculus that allows us to model expressions that involve perspectives. Our extension is derived from previous work in the semantics of programming languages aimed at providing a mathematical characterization of computations that produce some kind of *side effect* [Moggi, 1989], and is based on the notion of *monads* [Moggi, 1989, Wadler, 1992a, 1994, 1995], which we have used in a number of previous papers to model analyses of natural language meaning [Giorgolo and Asudeh, 2011, 2012b,a, 2014a,b], based on the pioneering work of Shan [2001]; see also Charlow [2014].[11] Monads are a construction in category theory that defines a canonical way to map a set of objects and functions that we may consider simple in some sense into a more complex space of objects and functions. They have been successfully used in the semantics of programming languages to characterise certain classes of computation [Moggi, 1989, Wadler, 1992a, 1995]; see Giorgolo and Asudeh [2014a] for some further discussion.

In the present case 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.[12] Effectively we will construct a kind

[11] Monads are related to *continuations* [Wadler, 1994], which are the formal tool used in a rich body of work by Chris Barker and Ken Shan (see Barker and Shan 2014 and references to their antecedent work therein). Charlow [2014] provides a particularly insightful study of the interplay of monads and continuations as applied to natural language semantics. There is also a relationship between our work and recent work by Jim Pryor on *mental graphs* [Pryor, 2015]. It would be interesting to explore how our monad-based approach and its results relate to the work of Barker and Shan, Charlow, and Pryor.

[12] Our indices should not be confused with those of Fiengo and May [1998] or with the kinds of indices that are commonly used in Logical Form semantics [Heim and Kratzer, 1998], as discussed in section 3.3.1,

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 (3.20) or the similar following example can be assigned non-contradictory readings:[13]

(3.36) 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.[14] 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.

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 $i \to \tau$. The functor $\Diamond_2$ maps any function $f : \tau \to \delta$ to a function $f' : (i \to \tau) \to i \to \delta$. $\Diamond_2$ corresponds to function composition:

$$\Diamond_2(f) = \lambda g.\lambda i.f(g(i)) \tag{3.37}$$

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 $i \to \tau$.

η (pronounced 'unit') is a polymorphic function that maps inhabitants of a type τ to inhabitants of its image under $\Diamond$, formally $\eta : \forall\tau.\tau \to \Diamond\tau$. 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:

$$\eta(x) = \lambda i.x \tag{3.38}$$

or in binding theory [Büring, 2005]. We return to a comparison of our indices to those of Fiengo and May in section 3.5.

[13] 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 (3.36) could have been said sincerely by Green in that context. `http://en.wikipedia.org/wiki/Reza_Aslan`

[14] The nature of these entities is discussed in section 3.4.

$\star$ (pronounced '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.[15] 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 (3.39).

$$a \star f = \lambda i.f(a(i))(i) \tag{3.39}$$

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 following three axioms:

$$\eta(x) \star f = f(x) \tag{3.40}$$

$$m \star \eta = m \tag{3.41}$$

$$(m \star f) \star g = m \star \lambda x.(f(x) \star g) \tag{3.42}$$

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 (3.38) and (3.39). These reductions are implicit in our analyses in section 3.4.

Composition

For semantic composition we use a logical calculus adapted for the linear case[16] from the one introduced by Benton et al. [1998]. The calculus is based on a language with two connectives corresponding to our type constructors: forward (/) and backward (\), binary connectives, that correspond to functional types, and $\Diamond$, a unary connective, that represents monadic types.

The logical calculus is described by the proof rules in Figure 3.1, repeated from Figure 1.3 in Lecture 1.[17] The rules come annotated with lambda terms that characterize the Curry-Howard correspondence between proofs and meaning terms [Curry and Feys, 1958, Howard, 1980, de Groote, 1995]. The exact nature of the underlying syntactic formalism is not strictly relevant, so long as it can deliver resources for semantic composition.

[15] 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.

[16] Various researchers have argued for linearity as a property of composition in natural language semantics (Moortgat 1999, Moortgat 2011, Asudeh 2012, among others). Asudeh [2012] discusses it under the rubric of 'resource sensitivity'.

[17] We can prove that the Cut rule is admissible, therefore the calculus becomes an effective (although inefficient) way of computing the meaning of a linguistic expression.

$$\frac{}{x : A \vdash x : A}\, id \qquad \frac{\Gamma \vdash B \qquad B, \Delta \vdash C}{\Gamma, \Delta \vdash C}\, Cut$$

$$\frac{x : A, \Gamma \vdash t : B}{\Gamma \vdash \lambda x.t : A \backslash B}\, \backslash R \qquad \frac{\Delta \vdash t : A \qquad \Gamma, x : B, \Theta \vdash u : C}{\Gamma, \Delta, y : A \backslash B, \Theta \vdash u[y(t)/x] : C}\, \backslash L$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : B / A}\, / R \qquad \frac{\Delta \vdash t : A \qquad \Gamma, x : B, \Theta \vdash u : C}{\Gamma, y : B / A, \Delta, \Theta \vdash u[y(t)/x] : C}\, / L$$

$$\frac{\Gamma \vdash t : A \qquad \Delta \vdash u : B}{\Gamma, \Delta \vdash \langle t, u \rangle : A \otimes B} \otimes R \qquad \frac{\Gamma, x : A, y : B, \Delta \vdash t : C}{\Gamma, \langle u, v \rangle : A \otimes B, \Delta \vdash t[x/u, y/v] : C} \otimes L$$

$$\frac{\Gamma \vdash x : A}{\Gamma \vdash \eta(x) : \Diamond A}\, \Diamond R \qquad \frac{\Gamma, x : A, \Delta \vdash t : \Diamond B}{\Gamma, y : \Diamond A, \Delta \vdash y \star \lambda x.t : \Diamond B}\, \Diamond L$$

Figure 3.1 Type-logical grammar with monadic modality as sequent calculus.

The axiom rule, the Cut rule and the rules for the introduction to the right and the left of the slashes are familiar from standard Lambek calculus, but the rules for the monadic connective $\Diamond$ deserve some comment. The rule for introduction to the right of $\Diamond$, when read from top to bottom, basically states that if we have a derivation that proves that from a set of resources/hypotheses Γ we can derive a resource/proposition A, then we are also able to derive from the same set of resources/hypotheses the same resource/proposition "encapsulated" into a monadic layer. If we interpret $\Diamond$ as representing an operation that allows us to lift a value into a new space of values, the rule allows us to lift a value into this new space. If we interpret the monadic connective in terms of computations, the rule states that we can create a computation that has as a result the resource that we started with.

The nature of this computation is perhaps better understood if we look at the lambda term associated with the rule. The new computation is produced by wrapping the original result with the unit η. Remember that the unit acts as a lifting operator that actually does not do anything besides taking the original value to a different value space. In this sense the right-hand rule for $\Diamond$ is safe as it does not introduce any additional information in the system.

The rule for introduction of $\Diamond$ to the left is slightly more complicated. First of all notice that, despite being a rule governing the behaviour of a connective on the left-hand side of the turnstyle, the rule imposes some restrictions on what is on the right-hand side of the sequent, namely that the resulting resource/proposition is a monadic one (but notice that with the right-hand rule for $\Diamond$ we can always construct a "dummy" monadic resource). The rule thus controls how we can introduce a monadic resource in the set of hypotheses. Again reading it top-down, if we have a proof that from a set of resources that contain a certain resource A we can produce a monadic resource $\Diamond B$, then we can produce the same resource if we take the same set of resources and replace A with a monadic resource that encapsulates A.

Again, looking at the lambda term associated with the rule gives us an idea of how

this replacement is performed. We assume that the original proof is encoded in the term t associated with $\Diamond B$, and that the original resource A contributes a term to t that we represent with variable x. Now, if the lambda term associated with the monadic resource $\Diamond A$ is y, we can extract from it the encapsulated value that corresponds to A and give it the name x so that it is substituted in t for the original x. In terms of proof search (i.e. looking at the rule bottom-up), the left rule ensures that a monadic resource on the left which is not consumed directly by some negative context (such as those set up on the left-hand side of an implication) is matched by a monadic layer in the result formula. This is because we do not in general have a function that maps from the monadic layer to the non-monadic one. However, for the specific monad that we are interested in (and many others) we can define such a function; in our case the function represents taking the default perspective, that of the speaker.

In the formalism section of the notes for Lecture 1, we presented derivations that exemplify some of the main derivation schemata at the core of our analysis. You can also test the relevant readings with the Day 3 grammar on the course website.

A key advantage of the monadic approach is that we are not forced to generalize all lexical entries to the "worst case", or richest type (as in, e.g., standard Montague Semantics or the semantics sketched in section 3.3.1). With the logical setup we have just described we can freely mix monadic and non-monadic resources. For example, we can combine a pure version of a binary function with arguments that are either pure or monadic, as the following are all provable theorems in our logic.

$$(C/B)/A, A, B \vdash \Diamond C \tag{3.43}$$
$$(C/B)/A, \Diamond A, B \vdash \Diamond C \tag{3.44}$$
$$(C/B)/A, A, \Diamond B \vdash \Diamond C \tag{3.45}$$
$$(C/B)/A, \Diamond A, \Diamond B \vdash \Diamond C \tag{3.46}$$

In contrast, the following is not a theorem in the logic:

$$(C/B)/A, I/A, I/B \nvdash C/I \tag{3.47}$$

In short, if we were to instead simply lift the type of the lexical resources whose interpretation may be dependent on a specific perspective, we would be forced to lift all linguistic expressions that may combine with them, thus generalizing to the worst case. We do not have to do this, given our logic.

The monadic machinery also achieves a higher level of compositionality. In principle we could directly encode our monad using the $\rightarrow$ type constructor, with, e.g. forward slash, on the logical side. However this alternative encoding wouldn't have the same deductive properties. Compare the pattern of inferences we have for the monadic type, in (3.43)–(3.46), with the corresponding pattern for the simple type:

$$(C/B)/A, A, B \vdash C \tag{3.48}$$
$$(C/B)/A, I/A, B \vdash C/I \tag{3.49}$$
$$(C/B)/A, A, I/B \vdash C/I \tag{3.50}$$
$$(C/B)/A, I/A, I/B \vdash (C/I)/I \tag{3.51}$$

In the case of the simple types (3.48)–(3.51), the final formula we derive depends in some non-trivial way on the entire collection of resources on the left-hand side of the sequent. In contrast, in the case of the monadic types (3.43)–(3.46), the same result type can be derived for all configurations. What is important is that we can predict the final formula without having to consider the entire set of resources available. This shows that the compositionality of our monadic approach cannot be equivalently recapitulated in a simple type theory.

3.4 Analysis

We will exemplify our approach with analyses of a selection of the examples discussed above, repeated here for convenience:

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

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

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

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

Example (3.54) is to be understood given the context that MJ does not know Peter Parker's secret and example (3.55) is to be understood in a Capgras context. The starting point for our analysis of these examples is the lexicon in Table 3.2. The lexicon represents the linguistic knowledge of the speaker, including her knowledge of other individuals' grammars.[18]

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 $\Diamond e$.[19] 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,[20] while Mary Jane and Dr. Octopus consider Spider-Man to be a different entity from Peter Parker.

[18]We have simplified some entries in Table 3.2 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*.

[19]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 3.5 that our solution is more general than this.

[20]See footnote 9 for some further clarification of this point.

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 3.2 Speaker's lexicon.

We assume an internalist semantics such that sentences are interpreted in a model in which all entities are mental entities, i.e. that there is no direct reference to entities in the world, but only to mental representations. This stance is more consistent with standard generative views about the nature of language [Chomsky, 2000, Larson and Segal, 1995, Ludlow, 2003], than the externalist view in the philosophy of language [Lau and Deutsch, 2014], which is often adopted tacitly in linguistic semantics. Entities are therefore relativized with respect to the individual that mentally represents them, where entities that the speaker believes to be non-contentious are always relativized according to the speaker. This allows us to represent the fact that different individuals may have distinct equivalencies between entities. For example, Kim in our model does not equate the evening star and the morning star, but the speaker equates them with each other and with Venus. Therefore, the speaker's lexicon in Table 3.2 represents the fact that the speaker's epistemic model includes what the speaker knows about other individuals' models, e.g. that Kim has a distinct denotation (from the speaker) for Hesperus, that Mary Jane has a distinct representation for Spider-Man, that Kim has a distinct representation for Sandy, etc.[21]

With respect to names, then, we adopt a certain kind of Millian stance, which we might call 'Representational Millianism': names refer directly, not by virtue of descriptions, but by virtue of direct reference to mental representations. A standard objection to this kind of move is that it makes semantics inherently "subjective", as discussed notably by Haas-Spohn [1995], which in turn makes communication impossible. This is too strong a conclusion: Representational Millianism makes communication about the same entity less direct, but does not render it impossible. Two agents succeed in communicating about the same thing in the world if the mental representations that their grammars makes reference to are representations of that same thing. In short, the argument against Representational Millianism is a simplicity argument (it posits mental representations that Naive Millianism does without), but simplicity arguments only cut ice if all else is held constant. However, we have argued that our account captures cases that Naive Millianism does not (see section 3.2.1). Moreover, mental representations are independently motivated throughout the cognitive sciences [Pitt, 2013]. Lastly, although some philosophers and linguists consider the externalism–internalism debate "settled" by arguments like those of Putnam [1975] and Burge [1979], this is a problematic conclusion. First, there is a large body of subsequent literature responding to these arguments and, for what it's worth, only a narrow majority of philosophers currently self-identify as externalists [Lau and Deutsch, 2014]. Second, the classic arguments are based on thought experiments, but the question is really an empirical one about how a natural object (the brain) works: the issue could never be settled entirely by thought experiments alone [Cummins, 1991, Chomsky, 1995].

We should stress that this internalist stance is not a *necessary* stance for our formal theory, but we think it is a *sensible* one, despite its potentially controversial nature. With respect to our formal theory, it does not matter *what* the model for interpretation is a model *of*, whether mental representations or reality. However, the representational layer that internalism offers us provides a way to make sense of the notion of distinct

[21] The speaker's Kim-denotation of *Sandy* is then not plausibly Kim's actual denotation — a mental representation that would seem privileged to Kim — but rather the speaker's representation of that representation.

denotations, which is especially relevant to the Capgras and Aslan cases. For example, in the Reza Aslan case, Aslan and Lauren Green were not in disagreement about which actual historical figure they were referring to, but rather about which properties that *very same person* had (see footnote 13 above for relevant details of this incident).

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.:

$$\kappa : e \to i \tag{3.56}$$

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

$$\kappa(\mathbf{r}_\sigma) = \mathtt{r} \tag{3.57}$$
$$\kappa(\mathbf{k}_\sigma) = \mathtt{k} \tag{3.58}$$
$$\kappa(\mathbf{o}_\sigma) = \mathtt{o} \tag{3.59}$$
$$\kappa(\mathbf{mj}_\sigma) = \mathtt{mj} \tag{3.60}$$

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.

Figure 3.2 shows the four non-equivalent readings that we derive in our system for example (3.52), repeated here as (3.62).[22]

(3.62) 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.[23]

[22]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:

$$m \star \lambda x.n \star \lambda y.p = n \star \lambda y.m \star \lambda x.p \tag{3.61}$$

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

[23]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.

Reading (3.67) 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 (3.63), 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.

$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{es}_\mathbf{k} = \mathbf{ms}_\mathbf{k}) \tag{3.63}$$

In the case of readings (3.68) and (3.69), 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 (3.64) and (3.65).

$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{ms}_\mathbf{k}) \tag{3.64}$$
$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{es}_\mathbf{k}) \tag{3.65}$$

Finally for reading (3.70) 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.

$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{v}_\sigma = \mathbf{v}_\sigma) \tag{3.66}$$

$$\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Hesperus}\rrbracket \star \lambda x.\,\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.67}$$

$$\llbracket\text{Hesperus}\rrbracket \star \lambda x.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.68}$$

$$\llbracket\text{Phosphorus}\rrbracket \star \lambda x.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Hesperus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(y)(x)))(\llbracket\text{Kim}\rrbracket))) \tag{3.69}$$

$$\llbracket\text{Hesperus}\rrbracket \star \lambda x.\,\llbracket\text{Phosphorus}\rrbracket \star \lambda y.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.70}$$

Figure 3.2 Non-equivalent readings for *Kim doesn't believe Hesperus is Phosphorus.*

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, our analysis of a sentence like (3.62) does not specify what the actual case is with respect to the mind-external reality of any of the readings. 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 (3.62), 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.[24]

If we consider a case like sentence (3.53), repeated in (3.71), 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*).

(3.71) #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:

$$\begin{array}{l}[\![\text{Spider-Man}]\!] \star \lambda x.\, [\![\text{Spider-Man}]\!] \star \\ \lambda y.\eta([\![\text{but}]\!]\,([\![\text{punch}]\!]\,([\![\text{Dr. Octopus}]\!])(x))([\![\text{not}]\!]\,([\![\text{punch}]\!]\,([\![\text{Dr. Octopus}]\!])(y))))\end{array} \tag{3.72}$$

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:

$$\mathbf{punch}(\mathbf{o}_\sigma, \mathbf{pp}_\sigma) \wedge \neg\mathbf{punch}(\mathbf{o}_\sigma, \mathbf{pp}_\sigma) \tag{3.73}$$

In contrast a verb like *love* is defined in the lexicon in Table 3.2 as possibly changing the interpretation perspective about its object to that of its subject. Therefore in the case of a sentence like (3.54), repeated in (3.74), we expect one reading where the potentially contentious name *Spider-Man* is interpreted according to the subject of *love*, Mary Jane.

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

This is in fact the result we obtain. Figure 3.3 reports the two readings that our framework generates for (3.74).

Reading (3.77), corresponds to the non-contradictory interpretation of sentence (3.74), where Spider-Man is interpreted according to Mary Jane's perspective and therefore is assigned an entity different from Peter Parker:

$$\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \wedge \neg\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{sm}_{\mathtt{mj}}) \tag{3.75}$$

Reading (3.78) instead generates unsatisfiable truth conditions, as Spider-Man is identified with Peter Parker according to the speaker's interpretation:

$$\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \wedge \neg\mathbf{love}(\mathbf{mj}_\sigma, \mathbf{pp}_\sigma) \tag{3.76}$$

[24] 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.

$$\eta(\llbracket\text{but}\rrbracket\,(\llbracket\text{love}\rrbracket\,(\eta(\llbracket\text{Peter Parker}\rrbracket))(\llbracket\text{Mary Jane}\rrbracket))\\ (\llbracket\text{not}\rrbracket\,(\llbracket\text{love}\rrbracket\,(\llbracket\text{Spider-Man}\rrbracket)(\llbracket\text{Mary Jane}\rrbracket)))) \tag{3.77}$$

$$\llbracket\text{Spider-Man}\rrbracket \star \lambda x.\eta(\llbracket\text{but}\rrbracket\,(\llbracket\text{love}\rrbracket\,(\eta(\llbracket\text{Peter Parker}\rrbracket))(\llbracket\text{Mary Jane}\rrbracket))\\ (\llbracket\text{not}\rrbracket\,(\llbracket\text{love}\rrbracket\,(\eta(x))(\llbracket\text{Mary Jane}\rrbracket)))) \tag{3.78}$$

Figure 3.3 Non-equivalent readings for *Mary Jane loves Peter Parker but she doesn't love Spider-Man.*

Our last example, the Capgras example (3.55), repeated here as (3.79), 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*.

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

Our system generates three non-equivalent readings, reported here in Figure 3.4.[25]

Reading (3.83) and (3.84) 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 (3.85) corresponds to the interpretation that assigns two different referents to the two instances of the name *Sandy*, producing the truth conditions in (3.80) which are satisfiable in a suitable model.

$$\neg\mathbf{B}(\mathbf{k}_\sigma, \mathbf{s}_\sigma = \mathbf{imp}_\mathbf{k}) \tag{3.80}$$

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

The analysis of the Aslan/Jesus example (3.36), repeated in (3.81), is equivalent; the non-contradictory reading is shown in (3.82).

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

$$\neg\mathbf{B}(\mathbf{r}_\sigma, \mathbf{j}_\sigma = \mathbf{j}_\mathbf{r}) \tag{3.82}$$

There are again three non-equivalent readings, including the one above, which are just those in Figure 3.4, with ⟦Sandy⟧ replaced by ⟦Jesus⟧ and ⟦Kim⟧ replaced by ⟦Reza⟧.

[25] Again the system generates six non-equivalent readings (see footnote 22), which are further reduced in this case as we have the same linguistic term appearing twice and combined with a commutative predicate (*is*).

$$\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Sandy}\rrbracket \star \lambda x.\,\llbracket\text{Sandy}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.83}$$

$$\llbracket\text{Sandy}\rrbracket \star \lambda x.\,\llbracket\text{Sandy}\rrbracket \star \lambda y.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.84}$$

$$\llbracket\text{Sandy}\rrbracket \star \lambda x.\eta(\llbracket\text{not}\rrbracket\,(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Sandy}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket\,(x)(y)))(\llbracket\text{Kim}\rrbracket))) \tag{3.85}$$

Figure 3.4 Non-equivalent readings for *Kim doesn't believe Sandy is Sandy.*

3.5 Comparison with Previous Approaches

Our approach seems superficially similar to Fiengo and May's, since both approaches are concerned with substitutability and use indices, but it is in fact quite distinct. First, the use of indices here resides entirely in the model. We do not require that the expression '$[_{\text{NP}}$ Sandy]' bear an index in order to address the puzzles above.[26]

Second, we are not forced to agree with Fiengo and May [1998, 381] that 'there is no value associated with the name 'Max' qua lexical item', despite agreeing with their claim that 'if we are to determine the identity conditions for words . . . we should determine the identity conditions on words as *they appear in the lexicon of an individual*.' (emphasis in the original; Fiengo and May 1998, 379).[27] Fiengo and May thus claim that the name *Max* in the lexicon does not refer, only the syntactic structure $[_{\text{NP}_i}$ Max] refers. This claim is not unreasonable, although it's certainly hard to see how to test it empirically. However, it does not seem to fit our folk understanding of names, which do seem referential in their own right, as words. Now, we are perfectly comfortable with folk understanding giving way to theory, but if two theories are equal in all other respects, it would seem to us slightly capricious, if not outright perverse, to pick the one that rejects folk understanding of its phenomena. On our theory, a name like *Max* does have its (range of) reference determined in the (speaker's) lexicon. There is some sense, then, in which our theory is more natural.

Third, it is well known that these puzzles are not just about names, but also about natural kinds and other predicates. It is hard to see how the kinds of indices that Fiengo and May [1998] use could be generalized to cover such cases; they seem too narrowly conceived to do the job. In contrast, a virtue of our analysis is that we can apply it to not just names and referring expressions, but to any natural language expression that may have different perspectival interpretations. This means that we can extend our analysis to other cases, such as the standard examples involving synonymous natural kind terms like *groundhog* and *woodchuck* (see, e.g., Fox and Lappin 2005) or *furze* and *gorse* [Kripke, 1979] or of synonymous verbs, such as *photocopy* and *xerox* [Larson and Ludlow, 1993].

As an illustration, consider the following example:

[26] This does not mean that there could not be other, independent reasons for expressions to bear indices, although we are most sympathetic to the spirit of David Beaver's question, 'What do those little numbers mean, and who put them there anyway?' [Beaver, 1999].

[27] The full quote has 'in particular, names' where we have the ellipsis, but we are not in fact convinced that names are special in this regard, for reasons that will become clear shortly.

(3.86) Elena thinks Flipper is a dolphin, but she doesn't think he is a marine mammal.

Suppose Elena thinks that Flipper is a dolphin and Hoover is a seal, but she thinks only Hoover is a marine mammal; i.e. she thinks seals are marine mammals, but dolphins are not. Suppose also that the speaker and Elena are in agreement about which entities the names *Flipper* and *Hoover* refer to, so the names are not controversial. Table 3.3 sketches (the relevant part of) the lexicon for the speaker of (3.86). We do not mean to imply that this extension of our approach is trivial, since matters of compositionality of, e.g. *marine mammal*, have not been addressed here, but the extension is at least a natural candidate for further exploration.

WORD	DENOTATION	TYPE
dolphin	$\{\mathbf{flipper}_\sigma\}$	$e \to t$
seal	$\{\mathbf{hoover}_\sigma\}$	$e \to t$
marine mammal	$\lambda i.\begin{cases}\{\mathbf{hoover}_\sigma\} & \text{if } i = \mathsf{e},\\ \{\mathbf{flipper}_\sigma, \mathbf{hoover}_\sigma\} & \text{if } i = \sigma\end{cases}$	$\Diamond(e \to t)$

Table 3.3 (Relevant portion of) speaker's lexicon for (3.86)

Lastly, let us return to the matter of traditional approaches to substitutability/opacity based on *de re/de dicto* ambiguities derived from differential compositional scopings. In section 3.2.3, we discussed certain problems that such an approach may have for ordinary proper names in Capgras and Indiana Pi Bill examples, repeated here:

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

(3.88) Dr. Goodwin doesn't believe π is π.

Here we outline some further issues for standard scope approaches.

First, substitutability puzzles in simple sentences, as discussed in section 3.2.1, pose a prima facie problem, since there does not seem to be a relevant scope operator present in the case of the composition of a simple predicate like *love* or *murder* with its object argument. Of course, many linguists would be perfectly willing to postulate null operators in such cases, but it is not clear that this derives the right result. That would be tantamount to treating *love* or *murder* as an 'opaque transitive verb' like *owe*, but the former have existential entailments that the latter lack [Zimmermann, 2006]:

(3.89) Frodo owes Sam a horse.
$\not\to$ There is a horse that Frodo owes Sam.

(3.90) Saruman murdered a horse.
$\to$ There was a horse that Saruman murdered.

(3.91) Princess Caroline loves a horse.
$\to$ There is a horse that Princess Caroline loves.

An account that postulates a hidden operator for *murder* or *love* would have to explain this contrast.

Second, even in the case of embedded contexts, which offer a scopal operator in composition, in the form of a modal or propositional attitude verb, a scope-based *de re/de dicto* approach faces some problems, at least in the case of our most challenging examples, such as the Indiana Pi Bill example, the Capgras example, and the Aslan/Jesus example (repeated and discussed further below). To try to explain the two readings in the context of a standard possible worlds semantics, we could take the Capgras example (3.87) to be ambiguous with respect to a *de re/de dicto* reading. In the case of the *de dicto* reading (which corresponds to the unsatisfiable reading) the two names are evaluated under the scope of the doxastic operator *believe*, i.e. they both refer to the same entity that is assigned to the name *Sandy* in each accessible world.[28] Clearly this is always the case, and so (3.87) is not satisfiable. In the case of the *de re* reading, we assume that the two names are evaluated at different worlds that assign different referents to the two names. One of these two worlds will be the actual world and the other world one of the accessible worlds. The reading is satisfiable if the doxastic modality links the actual world with one in which the name *Sandy* refers to a different entity. Notice that for this analysis to work we need to assume that names behave like quantifiers with respect to scoping both over and under modal and propositional attitude operators, as discussed in section 3.2.3.

However, it has been argued that even if we model names as generalized quantifiers, they are scopeless [Zimmermann, 1993]. But this is problematic for a scopal approach to the Capgras example. It would predict that both instances of the name *Sandy* escape the scope of *believe*. The resulting reading would bind the quantified individual to the interpretation of *Sandy* in the actual world. This would capture only an unsatisfiable reading. To save the scopal approach, we would need to assume that names in fact are sometimes interpreted in the scope of operators.

Even assuming that we find a satisfactory solution for these inconsistencies, the standard *de re/de dicto* scopal approach cannot really capture the intuitions behind opacity in all contexts. Consider again our Aslan/Jesus example, repeated here:

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

Assume that there are two views about Jesus: Jesus as a divine being and Jesus as a non-divine, simply human being. Assume that Jesus is non-divine in the actual world and that Reza is an atheist; then the only possible reading is the unsatisfiable one, as the referent for Jesus will be the same in the actual world and all accessible Reza-belief-worlds. The problem is that the scopal approach assumes a single modal model, while in this case it seems that there are two doxastic models necessary, Reza's model and the speaker's model. In contrast, in our approach the relevant part of Reza's model is embedded inside the speaker's model and perspective indices indicate which interpretation belongs to Reza and which to the speaker.

[28] In order to put the discussion on firmer footing, we have adopted the language of a Hintikka-like analysis of propositional attitudes [Hintikka, 1969, 1975], since this view is currently influential in linguistics, but our points should follow without a substantive commitment to that sort of analysis, so long as the propositional attitude verb is assumed to provide a scope point, as is commonly assumed.

A subset of approaches to *de re* ascription are often collectively referred to as *descriptivist* approaches (among others, Kaplan 1968, Lewis 1979, Cresswell and von Stechow 1982, Percus and Sauerland 2003). These approaches share the assumption that what is believed is something more structured than a simple proposition, such as a pairing of an individual (the *res*, what the belief is about) and a property. As Lewis [1979, 521] puts it, '[S]ometimes property objects will do and propositional objects won't.' It does not strike us as likely that a descriptivist approach to *de re* ascription would give a satisfactory account of the Aslan/Jesus case (or the Capgras or Indiana Pi Bill cases), for similar reasons to the ones given in the previous paragraph. The property at stake is that of *being Jesus* (or *being Sandy* or *being* π). But this property contains the contentious name that is at issue. It seems this kind of approach equally needs some way of mixing models/perspectives.

Such approaches would also inherit some of the general problems of *de re* approaches. First, if a scope point were postulated for verbs like *love* or *murder*, their lack of opacity with respect to existential entailments would be unexplained, as discussed above. Moreover, there seems to be no relevant structural difference in the object that could explain the distinction between these verbs and, e.g. *punch* or *kill*. Lastly, such approaches would still be challenged by lack of substitutability in simple sentences, as discussed above.

We have thus far assumed that the relevant perspectives are the subject's and the speaker's, but this likely needs further refinement, in ways that seem straightforward for our system. In particular, we have treated the speaker's perspective as the default, but there may be circumstances in which the speaker is purposefully adopting an alternative perspective. This may give us some purchase on examples like the *Barbara Vine* example in Zimmermann [2005, 69], which hinges on a bookshop owner purposefully using a pseudonym of author Ruth Rendell to refer to only a subset of her books:

(3.93) I've read all of Barbara Vine's books.

Crucially, in this scenario, the bookshop owner and Z himself are both enlightened about the multiple identity. Zimmermann characterizes the bookshop owner as adopting the language of a novice, although he is in fact an expert. In our system this could be captured by the bookshop owner modifying his lexicon such that Barbara Vine is (at least temporarily) a contentious name, such that it refers to an entity that is distinct from Ruth Rendell so long as the index of evaluation is not the bookshop owner or Z. The details remain to be worked out, though.

3.6 Conclusion

We have offered a semantics of perspective that offers a solution to the substitutability puzzle in both simple and embedded contexts. Our solution extends to cases of distinct interpretations of tokens of the same name, which gives rise to a related puzzle. We exemplified this case with respect to simple identity cases, as in the Capgras, Indiana Pi Bill, and Aslan/Jesus examples. Our solution to these puzzles rests on an analysis in terms of a combination of different perspectives. We have claimed that the switch to a different perspective is triggered by specific lexical items, such as propositional attitude

verbs, but also verbs like *love* and *murder* which express some kind of perspective on the part of the subject of the verb towards its object, but which nevertheless cannot easily be argued to be opaque in their object position. The context switch is not obligatory, as witnessed by the multiple readings that the sentences discussed seem to have.

The formalization of our analysis is based on monads. The main idea of our formal implementation is that referring expressions that have a potential perspectival dependency can be implemented as functions from perspective indices to fully interpreted values. Similarly, the linguistic triggers for context switch are implemented in the lexicon as functions that can modify the interpretation context of their arguments. Monads allow us to freely combine these "enriched" meanings with standard ones, avoiding unilluminating generalization to the worst case. We have also seen how more traditional approaches, while capable of dealing with some of the examples we discuss, are not capable of providing a general explanation of the totality of observed phenomena. We briefly explored how our approach could be extended to other types of natural language expressions, such as anaphora, natural kind terms, nominal predicates, and verbs. Careful exploration of these extensions remains part of future work.

We have inevitably had to take positions on some issues that are far from settled, but we do not mean these positions themselves to be the main contribution of this paper. Rather, it seems to us that philosophers and linguists are in broad agreement that in some linguistic contexts there seems to be an "extra something" involved in interpreting names, and other expressions; we have made a formal proposal about what that extra something could be: perspectives.

Lecture 4

Uncertainty

4.1 Introduction[1]

Conjunction fallacies have been an active area of research in cognitive science for more than three decades now. The phenomenon was first discussed by Tversky and Kahneman [1983]. They noticed that in a task asking for ratings of the relative likelihoods of different events, the majority of the subjects consistently rated the likelihood of the conjunction of two events as higher than the likelihood of one of the conjoined events.

One of their examples is the well-known "Linda paradox". 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.

(4.1) 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)]

The context is obviously designed to bias towards the label *Feminist* and it is unproblematic and unsurprising that the relevant proposition is ranked most likely, but the result that the joint probability T&F is ranked higher than T is interesting, and constitutes an instance of *conjunction fallacy*: a conjunction of two propositions is reported by subjects to be more probably than the probability of one of the two propositions on its own.

This result disagrees with the rules of probability, as the probability of the conjunction of two events, being the intersection of the two events, cannot be higher than the likelihood of any of the two events, formally for any two events A and B:

$$P(A \text{ and } B) \leq P(A), P(B) \tag{4.2}$$

[1] This lecture is an edited version ofGiorgolo and Asudeh [2014b].

These results have been replicated by many studies that have investigated different ways in which this apparently fallacious response can be elicited Yates and Carlson [1986], Tentori et al. [2004].

The original explanation of these results by Tversky and Kahneman [1983] was in terms of representativeness. The authors claimed that the observed responses are due to the fact that subjects do not operate in terms of probabilistic reasoning, but rather use a *representativeness heuristic*. According to Tversky and Kahneman [1983], subjects check the degree of correspondence between the events and a certain model of reality and select those event that are closer to what the model predicts as being the more likely events. Representativeness tends to covary with frequency but not necessarily. A crucial point of Tversky and Kahneman [1983]'s analysis is that this heuristic operates on the conjunction as a whole, or as they put it:

> the judged probability (or representativeness) of a conjunction cannot be computed as a function (e.g., product, sum, minimum, weighted average) of the scale values of its constituents. [Tversky and Kahneman, 1983, p. 305]

This last property is rather problematic if we intend to integrate their observations with current linguistic theories of meaning composition. In fact, a core hypothesis of modern semantic theory is the *principle of compositionality* [Janssen, 2011, among many others], ultimately based on the philosophy of language of Frege [1891/1952]. This principle states that the meaning of a composite expression is determined by the meaning of its constituents. This is in clear contradiction with the analysis of Tversky and Kahneman. However compositionality has proven a very important tool in linguistic research, in theoretical semantics and in the philosophy of language, representing the semantic counterpart of linguistic productivity. Moreover compositionality seems to be a pervasive, and indeed quite successful, strategy in human cognition. Therefore we should be careful before discarding it.

Tversky and Kahneman's explanation has been challenged by a number of researchers. In particular Hertwig Hertwig and Gigerenzer [1999], Mellers et al. [2001], Hertwig et al. [2008] has proposed that conjunction fallacies are not real errors, but rather emerge because of the intrinsic ambiguity of linguistic operators such as the conjunction *and*. Another important contribution of this line of research has been the demonstration that in certain contexts conjunction fallacies do not arise that easily. This is particularly true of contexts in which subjects are in some way primed to reason in terms of frequencies; for instance, if subjects are presented with a scenario that explicitly introduces frequencies, or if subjects are required to express explicitly their judgements regarding the likelihood of different events in terms of numerical estimates rather than implicitly by ordering the events in terms of likelihood. A similar reduction in the number of fallacies measured can be obtained by "raising the stakes", for example by asking subjects to bet on their estimates.

In what follows we try to reconcile these different points of view on the basis of the data reported in the literature and our goal of combining the results with the notion of compositionality. Our model is similar in some way to the one proposed by Yates and Carlson [1986]. They argue that conjunction fallacies arise because subjects use different

strategies to evaluate the combination of multiple uncertain events. They model the strategy that generates the fallacies with what they call a "signed summation" model. The idea is to substitute probabilities with a different likelihood measure λ that takes values in the entire $\mathbb{R}$ line. Likely events are assigned a positive number as their likelihood measure, whereas unlikely events are assigned a negative one. According to this model the joint likelihood of two events is the sum of their likelihoods:

$$\lambda(A \text{ and } B) = \lambda(A) + \lambda(B) \tag{4.3}$$

Our model starts from the same assumption that there are multiple strategies that are employed by subjects when evaluating the likelihood of conjoined uncertain events. But instead of assuming that unrelated computational processes underpin the different strategies, we will show that it is possible to assume a single uniform process that computes the likelihood of the conjunction of two events from their two relative likelihoods, but that does so by using different but related representations of uncertainty, expressed as alternative algebraic structures. More specifically we will demonstrate how we can explain the results reported in the literature in terms of an algebraic structure known as a *semiring*. This mathematical object is at the heart of a specific instance of a mathematical structure known as a *monad*, which has independently been shown to be a good model for the composition of complex meanings in natural language semantics Shan [2001].

In our model, the observation made in the literature Hertwig and Gigerenzer [1999] that conjunction fallacies arise only under specific conditions and can be cancelled if other conditions are imposed is explained in terms of cognitive/computational economy. In fact the same computational structure, the monad, can be used together with a number of different underlying semirings, one of them being the probability semiring. We predict that, in general, if the subjects are presented with a task where there are "no stakes" they will base their judgement on the basis of a reasoning modeled using a representation corresponding to a semiring defined over a relatively simple set with generally simple operations. Using this strategy will generally lead subjects to make overconfident estimations, which tend not to correspond to the reality of things. If, on the other hand, subjects are forced to evaluate the consequences of their judgements, such as in the context of a gaming scenario, or if explicitly primed to think in frequentist terms, then we will observe a switch to a more complex representation, with properties that better approximate those of probability theory.

Crucially, in our model, logical operators such as *and* or *or* maintain their core logical meaning, while the probabilistic behaviour is determined by the context in which they operate. In this sense, with respect to Hertwig et al. [2008]'s analysis, we switch the ambiguity to the context rather than assuming that a word like *and* has multiple meanings. This is in fact a problem with Hertwig et al. [2008]'s analysis, which effectively treats *and* as ambiguous, i.e. truly polysemous.

Our approach is similar in spirit to the standard Gricean approach Grice [1975], although it differs in some important ways. Similarly to the Gricean view we assume that logical words like *and* have a non-ambiguous core logical meaning. In the standard Gricean approach the core meaning is further elaborated by "implicatures", additional default meanings that are not explicitly stated by the speaker. Crucially, implicatures are

determined by context and can be cancelled by explicitly stating that they do not hold. For instance, a numeral like *two* is usually understood to mean "exactly two", as in *Susy has two daughters*, but the implicature can be cancelled, as witnessed by examples like *Susy has two daughters, in fact she has three*.

Similarly, in our approach the core meaning of a word like *and* is its logical, non-probabilistic meaning. But instead of relying on implicatures defined by the context, we have a grammatical process (as part of the logic composition) that can lift this core meaning to the probabilistic level if other linguistic expressions introduce uncertainty in the interpretation process. For instance, if *and* is used to conjoin two uncertain events, then the conjunction itself will become uncertain. The role of context in our model is not to permit the derivation of related additional meanings, but instead it selects which specific mode of uncertainty is going to be used. Notice that this does not introduce a source of non-compositionality in our system: both modes of uncertainty — pure probability and the simpler approximation — function in a purely compositional fashion. The limited form of ambiguity that may be attributed to our system stems entirely from the choice of the form of ambiguity. In contrast, in Hertwig et al. [2008]'s analysis, *and* is treated as truly ambiguous, and its final interpretation requires context as an additional parameter.

As a final remark in this introduction, we want to stress that our approach is in its current form limited to the explanation of the conjunction of two events, as expressed by the conjunction of two propositions. The related phenomenon of "concept conjunction", as discussed for instance by Hampton [1988], shares a number of similarities with conjunction fallacies, but at the same time displays some crucial differences, at least at the linguistic level. For example, in the case of noun-noun compounds, such as *school furniture*, the denotation of the expression is not constructed by simply intersecting the denotations of its component parts, but the function involved operates in non-trivial ways on these meanings.[2] This is not the case for the conjunction of events, where the result is not an entirely novel type of event, but it is indeed the Boolean meet of the two sub-events.

4.2 Monads and Uncertainty

In what follows we will introduce two different mathematical structures, *semirings* and *monads*. For reason of space we will only sketch their definitions, trying to provide the reader with an intuitive understanding of their importance. The interested reader can find more details concerning semirings in any introductory text about algebra, and similarly monads are discussed in most recent textbooks about category theory.

A semiring is a set A with two distinguished elements 0 and 1 and equipped with two

[2] As a matter of fact, the operation involved may not be a function at all; see, e.g., Partee [1995].

binary operations $+$ and $\cdot$, satisfying the following axioms, for all x, y and $z \in A$

$$(x + y) + z = x + (y + z) \tag{4.4}$$
$$x + y = y + x \tag{4.5}$$
$$x + 0 = 0 + x = x \tag{4.6}$$
$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \tag{4.7}$$
$$x \cdot 1 = 1 \cdot x = x \tag{4.8}$$
$$x \cdot 0 = 0 \cdot x = 0 \tag{4.9}$$
$$x \cdot (y + z) = (x \cdot y) + (x \cdot z) \tag{4.10}$$
$$(x + y) \cdot z = (x \cdot z) + (y \cdot z) \tag{4.11}$$

In the case of the probability semiring, A is the real interval $[0, 1]$, with 0 and 1 representing the two units and $+$ and $\cdot$ defined in the usual way.[3]

Monads are a mathematical structure that arises in category theory in the study of the algebra of the functors from a category to itself Awodey [2010]. They have found successful application in the semantics of programming languages Moggi [1989], Wadler [1992a] and more recently they have been proposed in the linguistic literature as a method to model phenomena that challenge traditional formal models of meaning Shan [2001], Unger [2011], Giorgolo and Asudeh [2012b,a]. In this paper we use monads as a mathematical device to model uncertainty in natural language meaning, and how uncertainty is combined and propagated in composed expressions, such as the conjunction of two propositions or predicates.

Intuitively we can think of monads as ways to map between different types of objects, in particular as a way to map simple objects into more complex ones. Monads are special mappings because they represent the canonical way to map the simple space of objects to a more complex one in such a way that important properties that link the simple objects are preserved under the mapping.

The fact that probability distributions form a monad in the category of measurable spaces was an early discovery in category theory Lawvere [1962], Giry [1982]. Here we use a slightly different characterisation based on the use of monads in functional programming to model probabilistic calculi. In our case, the space of simple objects is represented by the collection of different semantic types of natural language expressions, e.g. expressions that refer to an individual, such as proper names, expressions that denote some truth about reality, such as propositions, and expressions that denote a collection of individuals, such as common nouns. The mappings between these objects are expressed by similar expressions that "bring" us from one type of expression to another: for example we take a predicate expressed by a verb as a way to map the individual denoted by the subject expression to a truth value (when uttering *John sleeps* the verb *sleeps* maps the individual John to truth if John is indeed asleep and to false otherwise). The probability monad lifts these simple objects to probability distributions over inhabitants of the types, and so the mappings are transformed so that they relate different probability distributions

[3] These definitions allow the result of summing two probability measures to be higher than 1, but our system normalizes all probabilities such that the sum is less than or equal to 1.

(in our previous example the predicate would map from the probability distribution corresponding to the denotation of the subject, one that possibly assigns the entire probability mass to the individual John, to the probability distribution over the truth values, effectively giving us an estimate of the likelihood of the event that John is asleep).

Formally our monad is defined by the triple $\langle P, \eta, \star \rangle$. P is a functor that operates in the way described above: it maps our semantic types to probability distributions over the inhabitants of the type and lifts the mappings between the types so that they operate between probability distributions. So for instance in the case of the type t of truth values the inhabitants of P are going to look like the distributions shown in figure 4.2

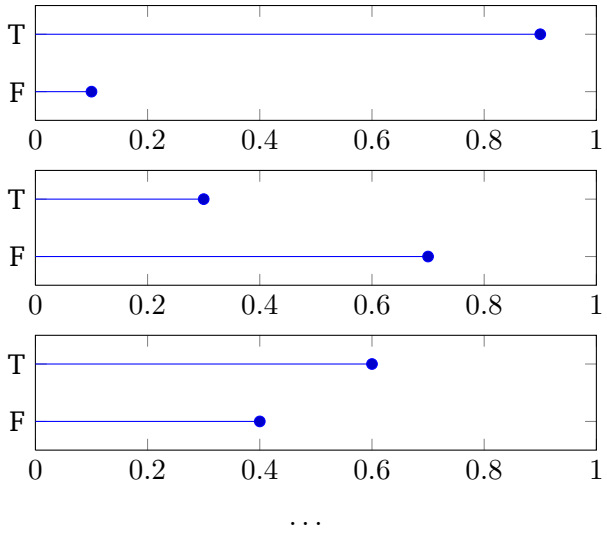


Figure 4.1 Some inhabitants of $P(t)$

η, pronounced "unit", is an operation that creates a special type of probability distribution that corresponds to a categorical judgement, i.e. assigning the entire probability mass to a single element of a semantic type. Basically this is a way to integrate certainty into our model of uncertainty. Formally we define it as follows (where we represent probability distributions as functions from types to the interval $[0, 1]$, or in general to any base set of a semiring):

$$\eta(x) := y \mapsto \begin{cases} 1 \text{ if } y = x \\ 0 \text{ otherwise} \end{cases} \tag{4.12}$$

where x represents an element of any semantic type (e.g. an individual, a truth value, a collection of individuals). For instance in the case of η_t, its application to T will result in the distribution shown in figure 4.2

The second operation, $\star$, pronounced "bind", is the way in which we combine uncertainty. Its definition is based on the definition of joint probability. Remember that in general the joint probability of two events is computed as follows:

$$P(A \text{ and } B) = P(A)P(B|A) \tag{4.13}$$

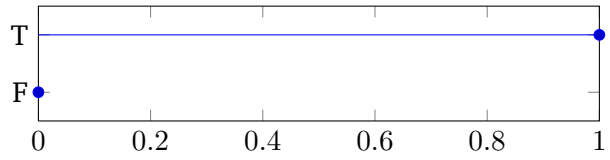


Figure 4.2 $\eta_t(T)$

It is important to notice that the joint probability we are discussing here is not the one we are trying to model in the case of a conjunction. The one discussed here is much more primitive and is at the core of the process that constructs the probability of an event from the linguistic elements that describe it. The joint probability that we will discuss in what follows arises as the byproduct of a large number of interactions described by the bind operation. There is also another difference between the way the bind operation is used and what we normally mean by joint probability. While the joint probability of two events gives us the likelihood of a third event (the occurrence of both atomic events), bind returns the probability distribution of what we can consider another atomic event. Bind is defined as in (4.14).

$$m \star k := y \mapsto \sum_{x \in A} m(x) \cdot k(x)(y) \tag{4.14}$$

where A is the set of elements measured by the probability distribution m. Bind takes two arguments, a probability distribution (m) and a function (k) from elements of the domain of the probability distribution m to probability distributions over elements of (possibly) another set. The resulting probability distribution is obtained by collecting all possible ways in which we can obtain the various results from this second set, and by scaling them ($\cdot$) using the likelihood that they emerge from the first distribution. The results are collected together using addition. Let's look at an example to better understand the workings of bind. Assume that the first argument of bind is represented by the probability distribution over some type A with three inhabitants, $\{a, b, c\}$, as represented in figure 4.2. The second argument of $\star$ is instead a function that maps each element of $\{a, b, c\}$ to

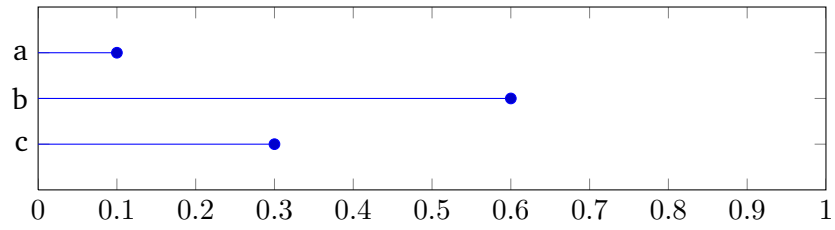


Figure 4.3 Example distribution over $\{a, b, c\}$.

a probability distribution over some type, let's say the type of truth values. For instance we could have the function in figure 4.2 The standard joint probability would give us the

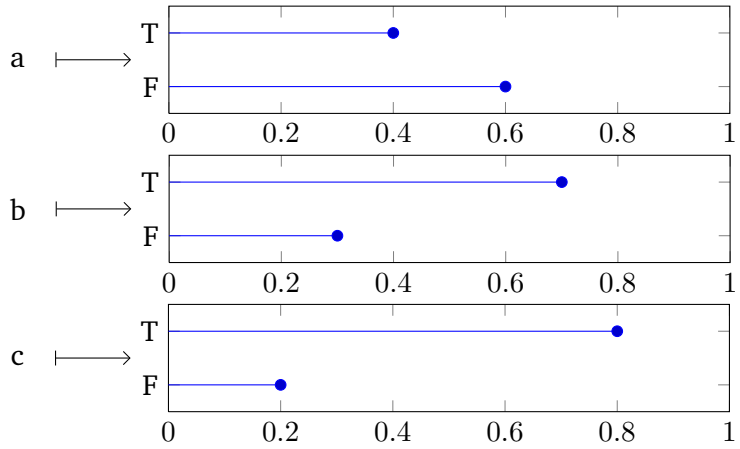


Figure 4.4 Example of a function from $\{a, b, c\}$ to probability distributions over truth values.

distribution shown in figure 4.2. But bind actually generates a distribution probability over the final type of its second argument, so the result we get is actually the one shown in figure 4.2.

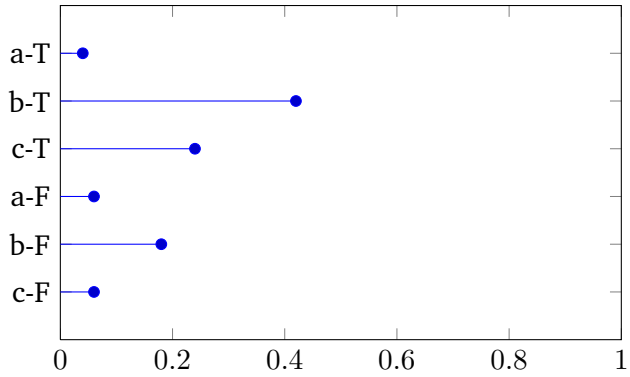


Figure 4.5 Pure joint probability for $\{a, b, c\}$ and $\{T, F\}$.

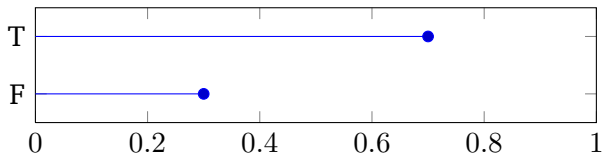


Figure 4.6 The result of applying bind to its two arguments

It is quite clear that despite the fact that we have been talking about a probability

monad, all operations involved in its definition are those we have described for a semiring. This means that we can use the same general structure we have discussed here and replace the meanings of 0, 1, $\cdot$, and $+$ with constants and operations defined for other semirings. This is precisely what we are going to do in the next section in which we explain how we can reproduce the results reported in the literature using our uncertainty monad.

4.3 Conjunction Fallacies, Compositionally

We show how we can reproduce the results reported in Yates and Carlson [1986] using our monadic infrastructure. We use their data as it is the only example in the literature where the relative likelihood of the atomic events in conjunctions has been (at least partially) controlled for. This gives us the possibility of deriving the overall likelihood of the conjoined event in a compositional fashion starting from the atomic events.

Likelihood of event A	Likelihood of event B	Observed rating
U	U	$P(A \text{ and } B) \leq P(A), P(B)$
U	L	$P(A) \leq P(A \text{ and } B) \leq P(B)$
L	L	$P(A), P(B) \leq P(A \text{ and } B)$

Table 4.1 Results reported by Yates and Carlson [1986]

The first step is to define a suitable base for our semiring. Yates and Carlson [1986] employ a discrete scale based on the general prediction made by their summation model. Their model does not take into consideration the extreme cases, i.e. impossible and certain events. We want to include them in our model as they are necessary in order to model what we know about the logical entailment behaviour of the word *and*. Tentori et al. [2004] showed that subjects that commit conjunction errors correctly apply the rules of logic. Therefore we will use a simple discrete set as the base for our semiring: {I(mpossible), U(unlikely), P(ossible), L(ikely), C(ertain)}. I and C correspond to 0 and 1 respectively. The only additional condition we need to impose so that I and C behave as boolean values is that for all x in our set $x + C = C$. There are sixteen possible well behaved semirings that we can define for this set.

The next step is to select those semirings that reproduce the results reported in Yates and Carlson [1986] and here summarised in table 4.1. To do so we have to explain how we expect the uncertainty attached to the two atomic events to propagate to the conjunction. This is described formally by the function in (4.15).[4]

$$\langle p, q \rangle \mapsto p \star (x \mapsto q \star (y \mapsto \eta(x \wedge y))) \tag{4.15}$$

This function takes a pair of probability distributions over truth values (describing how likely an event is realised in the real world) and constructs a new probability distribution over the results of conjoining the truth value of the first event with the one of the second.

[4]This function is actually generated in a completely compositional fashion by the grammatical system. For details on how such a system may work see Benton et al. [1998] for a logical perspective and Giorgolo and Asudeh [2012b] for a more linguistic one.

In this particular case there are three ways in which the final result can be false, and only one in which it turns out true (when both events are realized).

Using this schema we can evaluate how the sixteen semirings match the simplified results in table 4.1. It turns out that only one of the sixteen semirings satisfies the conditions induced by the data. We show the one semiring in table 4.2. Notice that this semiring is not homomorphic to the probability semiring, meaning that we cannot reproduce its behaviour using probability theory.

$+$	I	U	P	L	C
I	I	U	P	L	C
U	U	U	P	L	C
P	P	P	P	L	C
L	L	L	L	L	C
C	C	C	C	C	C

$\cdot$	I	U	P	L	C
I	I	I	I	I	I
U	I	U	U	U	U
P	I	U	P	P	P
L	I	U	P	L	L
C	I	U	P	L	C

Table 4.2 The one semiring.

4.4 One Process, Two Representations

As we have already explained, we assume that conjunction fallacies are true errors, in the sense that they lead to overestimation of the likelihood of events, but at the same time they are correct applications of a different strategy for computing the likelihood of composed events. Our explanation is given in terms of cognitive/computational economy. If there are no real stakes on the table, and therefore there is no incentive in using a safer but costlier computational system, subjects will employ a form of shortcut, represented in our model by a simpler semiring. If on the other hand, subjects are pushed to think about the consequences of their judgements, the more expensive solution is selected.

The semiring described in table 4.2 is undoubtedly simpler than the standard probability semiring. First of all it is based on a simpler base set. But it also has another important property: it is in fact possible to reconstruct the entire semiring on the basis of only one of the two operations and a simpler complement operation, $-$. If we define the complement as in table 4.3 (which seems to be most intuitive way to define it, as we just reverse the order of the relative likelihoods), then we can observe that for all x and y in our base set we have that $x \cdot y = -(-x + -y)$ or alternatively $x + y = -(-x \cdot -y)$.[5] This means that this particular encoding of uncertainty has much lower representational costs than other competing possibilities. Its symmetry makes it a particularly simple and efficient computational object.

$-$	I	U	P	L	C
	C	L	P	U	I

Table 4.3 Complement operation for the one semiring.

[5] It is interesting to note that these laws correspond to De Morgan's Laws in the semiring of truth values.

From a processing perspective this means that we can posit a single compositional process that computes the likelihood of two conjoined events in a way that is completely blind to the specific details of how uncertainty is encoded. We just require that the encoding satisfies the axioms of a semiring. Moreover we observe the same general process, here formalized in terms of monads, at play in other areas of natural language meaning Shan [2001], Unger [2011], Giorgolo and Asudeh [2012b,a]. The two encodings we have discussed are instead selected by the context under which the description of the conjoined event is evaluated. If there are no real risks and judgements have no real consequences, then we predict that subject will select a computationally and representationally cheap encoding (the one semiring), otherwise they will apply the rules of probability, which require a much higher degree of computation and are representationally more costly. In terms of psychological heuristics, our proposal is that conjunction fallacies involve the heuristic of *satisficing* Simon [1956], either in addition to or instead of the heuristic of *representativeness* Tversky and Kahneman [1983].

4.5 Conclusion

We have presented a uniform model that provides a uniform model for the interpretation of uncertain conjoined events. In particular, we have shown how it is possible to account for the presence of conjunction fallacies in judgements related to the likelihood of conjoined events, together with the correct application of the rules of probability theory. Our model starts from the idea that linguistically described events are interpreted starting from their linguistic description through a compositional process. This process is blind to the specific semantic content of the linguistic elements, in particular to the representation of the measure of uncertainty. We take that the context of evaluation has the effect of selecting one of two possible representations. One is computationally less expensive, but has the side effect of generating over-confident estimates for conjoined events. The other is computationally more expensive, but leads to judgements that are closer to the expected odds. We assume that the first representation is selected if there are no real consequences in case of an erroneous estimate, while the second is preferred if a mistake may have consequences.

The advantage of this approach over previous models is that it bridges an important hypothesis in the study of natural language semantics, compositionality, and a pervasive cognitive illusion such as the conjunction fallacy. Our model also explains why different strategies for evaluating uncertain events are selected, based on a simple computational criterion, and moreover the notion of cognitive/computational economy that we formally capture can be understood in light of the *satisficing heuristic*.

Our model also makes novel predictions that we plan to study in future work. First of all, the purely compositional nature of our model means that we can apply it, as is, to other cases that involve the combination of different events via logical operators, such as the cases of disjunction Carlson and Yates [1989] and implication. But our model predicts that similar effects should also be observable in cases where the conjunction of events is implicit, such as in the case of universally quantified sentences. One simple interpretation of such examples is in fact in terms of an iterated conjunction over the

domain of quantification. If we also observe conjunction fallacies in this cases, this should provide important evidence for a compositional model. But our model also suggests novel ways in which we can prime subjects to not commit reasoning fallacies related to uncertainty. If we are correct in assuming that subjects select the specific strategy for evaluating uncertain events on the basis of the possible repercussions of their choices, then we predict that we can force subjects to select a frequency based style of reasoning by simply introducing such consequences. Betting is of course one example, but other possibilities include some kind of emotional or social feedback to the judgements of subjects.

Lecture 5

Interactions, Connections and Wrapping Up

5.1 Introduction

Main ideas:

- We can represent different monads in the logic using different *modalities*.
- Each monad / modality represents a single linguistic phenomenon, isolated from other phenomena.
- The interaction between monads should be a by-product of the compositional process rather than being built-in in the lexicon.
- However in general the composition of two monads is not a monad; this is possibly reflected in the impossibility of deriving certain patterns in the logical side of the semantic derivation.
- But, if the two monads are connected by a distributive law then we should be able to make them "talk".

Open questions:

- What happens if we have three monads? (iterated distributive laws)
- What's the relation with the swapping patterns and monad composition? Do our derivations describe what could be achieved with a precomposed monad?
- Are the distributive laws we use unique? It is quite nice that the prediction of the complex phenomenon arise from the principled interaction of the simpler ones, but if we can show that the laws are unique the results are even more compelling.

Note: The attempt at playing with the interaction between different monads is a computational nightmare so the derivation are all done by hand. Using a sequent calculus to do that is in turn a nightmare, so instead we use a much lighter natural deduction format. The rules for the calculus are shown in table 5.1.

$$\frac{\begin{array}{c}[x:A]_1\\ \vdots\\ t:B\end{array}}{\lambda x(t):A \multimap B} \multimap I_i \qquad \frac{x:A \qquad f:A \multimap B}{f(x):B} \multimap E$$

$$\frac{x:A}{\eta(x):\Diamond A}\,\Diamond I \qquad \frac{m:\Diamond A \qquad \begin{array}{c}[x:A]_i\\ \vdots\\ n:\Diamond B\end{array}}{m \star \lambda x.n:\Diamond B}\,\Diamond E_i$$

Table 5.1 Natural deduction rules.

5.2 Conventional Implicature and Perspectives

As an example consider the following Capgras example:

(5.1) Kim doesn't believe goddamn Sandy is Sandy.

Intuitively the conventional implicature that the speaker has negative feelings towards Sandy should be interpreted outside the scope of "believe" (I doubt there is a genuine *de re* reading for this sentence but I should check if Chris has anything to say about it). Let's see what happens if we try to derive the meaning of this sentence.

As a first step we simply extend our logic, instead with a single monadic modality $\Diamond$, with a collection of modalities $\Diamond_i$, each one corresponding to a monad that models a different "complex" semantic phenomenon. The rules for the use of the modality on the left and the right of sequents are restricted to the same modalities:

$$\frac{\Gamma \vdash x:A}{\Gamma \vdash \eta_i(x):\Diamond_i A}\,\Diamond_i R \qquad \frac{\Gamma, x:A \vdash t:\Diamond_i B}{\Gamma, y:\Diamond_i A \vdash y \star_i \lambda x.t:\Diamond_i B}\,\Diamond_i L \tag{5.2}$$

The corresponding natural deduction rules are the following ones:

$$\frac{x:A}{\eta_i(x):\Diamond_i A}\,\Diamond_i I \qquad \frac{m:\Diamond_i A \qquad \begin{array}{c}[x:A]_n\\ \vdots\\ t:\Diamond_i B\end{array}}{m \star_i \lambda x.n:\Diamond_i B}\,\Diamond_i E_n \tag{5.3}$$

Each modality has its own unit and bind operations of course.

For our example we assume two modalities: $\Diamond_p$ for perspective taking, and $\Diamond_{ci}$ for conventional implicatures.

Starting with the complement we can show how the two monads are introduced in the derivation. We assume the lexical entries in table 5.2.

WORD	DENOTATION	TYPE	FORMULA
Kim	kim	e	k
not	$\lambda p.\neg p$	$t \to t$	$b \multimap b$
believe	$\lambda s.\lambda o.\lambda i.believe(s, o(s))$	$e \to \Diamond_p t \to \Diamond_p t$	$k \multimap \Diamond_p i \multimap \Diamond_p b$
Sandy	$\{kim \mapsto impostor, \sigma \mapsto sandy\}$	$\Diamond_p e$	$\Diamond_p s_1 / \Diamond_p s_2$
goddamn	$\lambda x.\langle x, bad(x)\rangle$	$e \to \Diamond_{ci} e$	$s_1 \multimap \Diamond_{ci} s_1$
is	$\lambda x.\lambda y.x = y$	$e \to e \to t$	$s_1 \multimap s_2 \multimap i$

Table 5.2 Lexicon.

For the NP "goddamn Sandy" we can derive a resource of type $\Diamond_p \Diamond_{ci} e$ as shown in figure 5.1. We can combine this resource with "is Sandy" to obtain a resource of type $\Diamond_p \Diamond_{ci} t$ as shown in figure 5.2 (this corresponds to the reading where both instances of Sandy are interpreted inside the scope of believe). Notice that this seems to be wrong type, as the conventional implicature material is made dependent on an interpretation index.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{y : s_1 \Rightarrow y : s_1}\,Id \quad \dfrac{}{x : \Diamond_{ci} s_1 \Rightarrow x : \Diamond_{ci} s_1}\,Id}{[\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1, y : s_1 \Rightarrow [\![\text{goddamn}]\!]\,(y) : \Diamond_{ci} s_1}\multimap L}{[\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1, y : s_1 \Rightarrow \eta_p([\![\text{goddamn}]\!]\,(y)) : \Diamond_p \Diamond_{ci} s_1}\Diamond_p R}{[\![\text{sandy}]\!] : \Diamond_p s_1, [\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1 \Rightarrow [\![\text{sandy}]\!] \star_p \lambda y.\eta_p([\![\text{goddamn}]\!]\,(y)) : \Diamond_p \Diamond_{ci} s_1}\Diamond_p L$$

Figure 5.1 Derivation for "goddamn Sandy"

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{v : \Diamond_{ci} s_1, [\![\text{is}]\!] : s_1 \multimap s_2 \multimap i, z : s_2 \Rightarrow v \star_{ci} \lambda w.\eta_{ci}([\![\text{is}]\!]\,(w)(z)) : \Diamond_{ci} i}\Diamond_{ci} L}{v : \Diamond_{ci} s_1, [\![\text{is}]\!] : s_1 \multimap s_2 \multimap i, z : s_2 \Rightarrow \eta_p(v \star_{ci} \lambda w.\eta_{ci}([\![\text{is}]\!]\,(w)(z))) : \Diamond_p(\Diamond_{ci} i)}\Diamond_p R}{[\![\text{sandy}]\!] : \Diamond_p s_2, v : \Diamond_{ci} s_1, [\![\text{is}]\!] : s_1 \multimap s_2 \multimap i \Rightarrow [\![\text{sandy}]\!] \star_p \lambda z.\eta_p(v \star_{ci} \lambda w.\eta_{ci}([\![\text{is}]\!]\,(w)(z))) : \Diamond_p(\Diamond_{ci} i)}\Diamond_p L}{[\![\text{gs}]\!] : \Diamond_p(\Diamond_{ci} s_1), [\![\text{sandy}]\!] : \Diamond_p s_2, [\![\text{is}]\!] : s_1 \multimap s_2 \multimap i \Rightarrow [\![\text{gs}]\!] \star_p \lambda v.\,[\![\text{sandy}]\!] \star_p \lambda z.\eta_p(v \star_{ci} \lambda w.\eta_{ci}([\![\text{is}]\!]\,(w)(z))) : \Diamond_p(\Diamond_{ci} i)}\Diamond_p L$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{w : s_1 \Rightarrow w : s_1}\,Id \quad \dfrac{\dfrac{}{z : s_2 \Rightarrow z : s_2}\,Id \quad \dfrac{}{y : i \Rightarrow y : i}\,Id}{x : s_2 \multimap i, z : s_2 \Rightarrow x(z) : i}\multimap L}{[\![\text{is}]\!] : s_1 \multimap s_2 \multimap i, z : s_2, w : s_1 \Rightarrow [\![\text{is}]\!]\,(w)(z) : i}\multimap L}{[\![\text{is}]\!] : s_1 \multimap s_2 \multimap i, z : s_2, w : s_1 \Rightarrow \eta_{ci}([\![\text{is}]\!]\,(w)(z)) : \Diamond_{ci} i}\Diamond_{ci} R$$

Figure 5.2 Derivation for "goddamn Sandy is Sandy" ($[\![\text{gs}]\!]$ stands for the meaning derived for "goddamn Sandy" as show in figure 5.1).

At this point we are stuck. "believe" can compose with its subject but then it expects a complement of type $\Diamond_p t$. Instead we have a complement of type $\Diamond_p \Diamond_{ci} t$. The closest thing we can do is to lift the type of the application of "believe" to "Kim" to the type $\Diamond_{ci} \Diamond_p t \to \Diamond_{ci} \Diamond_p t$, by just applying the second component of $\Diamond_{ci}$, but the argument still doesn't match as the monadic layers are swapped.

In fact, this is a known problem regarding the composition of monads. And a proposed solution to the issue of composing monads is exactly the definition of a *distributive law* between two monads to be composed, such that when composing two monads

$\langle m, \eta^m, \mu^m \rangle$ and $\langle n, \eta^n, \mu^n \rangle$ we need to define a natural transformation $\delta : nm \to mn$ (that of course has to satisfy a bunch of rules).

In our case we need to be able to swap the stack $\Diamond_{ci}\Diamond_p$ into $\Diamond_p\Diamond_{ci}$, that is we need to define a distributive law for the perspective monad when it's composed with the conventional implicature monad (the order is reversed). In other words, assuming that i is the type of interpretation indices and t^p the type of sets of propositions (the monoid of the conventional implicature monad) then we need a way to translate the type $i \to (\alpha \otimes t^p)$ into $(i \to \alpha) \otimes t^p$ for all types α. To be able to define such a translation we have to make explicit one assumption in our definition of the perspective monad. In fact, we implicitly assumed that the type of indices is non-empty and that there is a special default element of the type, σ, that we take to represent the perspective of the speaker. By specifying that our monad is the monad of functions from a fixed pointed set, whose "point" is σ then δ^p_{ci} can be defined as follows:

$$\delta^p_{ci}(f) = \langle \lambda i.\pi_1(f(i)), \pi_2(f(\sigma)) \rangle \tag{5.4}$$

The proof that this is a distributive law is in section 5.5.

Basically we are reusing the old monadic stack to generate the value, but the conventional implicatures are generated using the default interpretation index, σ. This matches the prediction that the side-issue comment should be attributed to speaker, independently of the context inside which it is evaluated.

This step can be captured by the following structural rules:

$$\frac{x : \Diamond_i\Diamond_j A}{\delta^i_j(x) : \Diamond_j\Diamond_i A}\, swap^i_j \qquad \frac{\Gamma, x : \Diamond_i\Diamond_j A \vdash t : B}{\Gamma, y : \Diamond_j\Diamond_i A \vdash t[\delta^j_i(x)/x] : B}\, swap_{ji}L \qquad \frac{\Gamma \vdash t : \Diamond_i\Diamond_j A}{\Gamma \vdash \delta^j_i(t) : \Diamond_j\Diamond_i A}\, swap_{ji}R \tag{5.5}$$

The rules are restricted by the existence of a distributive law between the modalities / monads involved.

In our case, we can also define the distributive law δ^{ci}_p (the monad used for conventional implicatures can be given a distributive law with any other monad), show here in (5.6) (where $\Diamond^2_p$ is the component of the perspective functor that maps between arrows).

$$\delta^{ci}_p(\langle x, m \rangle) = \Diamond^2_p(\lambda y.\langle y, m \rangle)(x) \tag{5.6}$$

With δ^p_{ci} we can see that we can compose the matrix clause with its complement: first we saturate the first argument of the verb applying it to its subject, then we lift the resulting

5.3 Natural Deduction Proofs

$$\eta_p(x) = \lambda i.x \tag{5.7}$$
$$m \star_p k = \lambda i.k(m(i))(i) \tag{5.8}$$
$$\eta_{ci}(x) = \langle x, \varepsilon \rangle \tag{5.9}$$
$$m \star_{ci} k = \langle \pi_1(k(\pi_1(m))), \pi_2(m) \cdot \pi_2(k(\pi_1(m))) \rangle \tag{5.10}$$

5.3.1 Unsatisfiable Reading 1

$$\dfrac{\dfrac{\dfrac{[\![\text{kim}]\!] : k \qquad [\![\text{believe}]\!] : k \multimap \Diamond_p i \multimap \Diamond_p b}{[\![\text{believe}]\!]\,([\![\text{kim}]\!]) : \Diamond_p i \multimap \Diamond_p b}\multimap E \qquad [x : \Diamond_p i]_1}{[\![\text{believe}]\!]\,([\![\text{kim}]\!])(x) : \Diamond_p b}\multimap E \qquad \dfrac{\dfrac{[\![\text{not}]\!] : b \multimap b \qquad [y : b]_2}{[\![\text{not}]\!]\,(y) : b}\multimap E}{\eta_p([\![\text{not}]\!]\,(y)) : \Diamond_p b}\Diamond_p I}{[\![\text{believe}]\!]\,([\![\text{kim}]\!])(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)) : \Diamond_p b}\Diamond_p E_2}{\eta_{ci}([\![\text{believe}]\!]\,([\![\text{kim}]\!])(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y))) : \Diamond_{ci}\Diamond_p b\ \boxed{\text{A}}}\Diamond_{ci} I$$

$$\dfrac{\dfrac{\boxed{\text{A}} \qquad [z : \Diamond_{ci}\Diamond_p i]_3}{z \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,([\![\text{kim}]\!])(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))}\Diamond_{ci} E_1}{\lambda z.z \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,([\![\text{kim}]\!])(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y))) : \Diamond_{ci}\Diamond_p i \multimap \Diamond_{ci}\Diamond_p b\ \boxed{\text{B}}}\multimap I_3$$

$$\dfrac{[\![\text{Sandy}]\!] : \Diamond_p s_1 \qquad \dfrac{\dfrac{[\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1 \qquad [w : s_1]_4}{[\![\text{goddamn}]\!]\,(w) : \Diamond_{ci} s_1}\multimap E}{\eta_p([\![\text{goddamn}]\!]\,(w)) : \Diamond_p\Diamond_{ci} s_1}\Diamond_p I}{[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) : \Diamond_p\Diamond_{ci} s_1\ \boxed{\text{C}}}\Diamond_p E_4$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\![\text{is}]\!] : s_1 \multimap s_2 \multimap i \qquad [v : s_1]_5}{[\![\text{is}]\!]\,(v) : s_2 \multimap i}\multimap E \qquad [u : s_2]_6}{[\![\text{is}]\!]\,(v)(u) : i}\multimap E}{\eta_{ci}([\![\text{is}]\!]\,(v)(u)) : \Diamond_{ci} i}\Diamond_{ci} I \qquad [t : \Diamond_{ci} s_1]_7}{t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u)) : \Diamond_{ci} i}\Diamond_{ci} E_5}{\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u))) : \Diamond_p\Diamond_{ci} i\ \boxed{\text{D}}}\Diamond_p I$$

$$\dfrac{\dfrac{\dfrac{\boxed{\text{C}} \qquad \boxed{\text{D}}}{[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u))) : \Diamond_p\Diamond_{ci} i}\Diamond_p E_7 \qquad [\![\text{Sandy}]\!] : \Diamond_p s_2}{[\![\text{Sandy}]\!] \star_p \lambda u.\,[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u))) : \Diamond_p\Diamond_{ci} i}\Diamond_p E_6}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda u.\,[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u)))) : \Diamond_{ci}\Diamond_p i\ \boxed{\text{E}}}\,swap^p_{ci}$$

$$\dfrac{\boxed{\text{B}} \qquad \boxed{\text{E}}}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda u.\,[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u)))) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y))) : \Diamond_{ci}\Diamond_p b}$$

$\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda u.\,[\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)(u)))) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\delta^p_{ci}(\lambda i.([\![\text{Sandy}]\!] \star_p \lambda w.\eta_p([\![\text{goddamn}]\!]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)([\![\text{Sandy}]\!]\,(i)))))(i)) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\delta^p_{ci}(\lambda i.(\eta_p([\![\text{goddamn}]\!]\,([\![\text{Sandy}]\!]\,(i))) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)([\![\text{Sandy}]\!]\,(i)))))(i)) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\delta^p_{ci}(\lambda i.\eta_p([\![\text{goddamn}]\!]\,([\![\text{Sandy}]\!]\,(i))) \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)([\![\text{Sandy}]\!]\,(i)))))(i)) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\delta^p_{ci}(\lambda i.([\![\text{goddamn}]\!]\,([\![\text{Sandy}]\!]\,(i))) \star_{ci} \lambda v.\eta_{ci}([\![\text{is}]\!]\,(v)([\![\text{Sandy}]\!]\,(i))))) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\delta^p_{ci}(\lambda i.\langle[\![\text{is}]\!]\,([\![\text{Sandy}]\!]\,(i))([\![\text{Sandy}]\!]\,(i)), bad([\![\text{Sandy(i)}]\!])\rangle) \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\langle \lambda i.\,[\![\text{is}]\!]\,([\![\text{Sandy}]\!]\,(i))([\![\text{Sandy}]\!]\,(i)), bad([\![\text{Sandy}]\!]\,(\sigma))\rangle \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\langle \lambda i.\,[\![\text{is}]\!]\,([\![\text{Sandy}]\!]\,(i))([\![\text{Sandy}]\!]\,(i)), bad(sandy)\rangle \star_{ci} \lambda x.\eta_{ci}([\![\text{believe}]\!]\,(kim)(x) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)))$

$\langle believe(kim, [\![\text{Sandy}]\!]\,(kim) = [\![\text{Sandy}]\!]\,(kim)) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)), bad(sandy)\rangle$

$\langle believe(kim, impostor = impostor) \star_p \lambda y.\eta_p([\![\text{not}]\!]\,(y)), bad(sandy)\rangle$

$\langle \lambda i.\neg believe(kim, impostor = impostor), bad(sandy)\rangle$

5.3.2 Unsatisfiable Reading 2

$$\dfrac{\dfrac{\dfrac{[\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1 \qquad [x : s_1]_1}{[\![\text{goddamn}]\!]\,(x) : \Diamond_{ci} s_1} \multimap E}{\eta_p([\![\text{goddamn}]\!]\,(x)) : \Diamond_p \Diamond_{ci} s_1} \Diamond_p I \qquad [\![\text{Sandy}]\!] : \Diamond_p s_1}{[\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x)) : \Diamond_p \Diamond_{ci} s_1} \Diamond_p E_1}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) : \Diamond_{ci} \Diamond_p s_1\ \boxed{\text{A}}}\ swap^p_{ci}$$

$$\dfrac{\dfrac{[\![\text{is}]\!] : s_1 \multimap s_2 \multimap i \qquad [y : s_1]_2}{[\![\text{is}]\!]\,(y) : s_2 \multimap i} \multimap E \qquad [z : s_2]_3}{[\![\text{is}]\!]\,(y)(z) : i} \multimap E}{\eta_p([\![\text{is}]\!]\,(y)(z)) : \Diamond_p i\ \boxed{\text{B}}} \Diamond_p I$$

$$\dfrac{\dfrac{[\![\text{Kim}]\!] : k \qquad [\![\text{believe}]\!] : k \multimap \Diamond_p i \multimap \Diamond_p b}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!]) : \Diamond_p i \multimap \Diamond_p b} \multimap E \qquad \boxed{\text{B}}}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) : \Diamond_p b} \multimap E \qquad \dfrac{\dfrac{[\![\text{not}]\!] : b \multimap b \qquad [v : b]_4}{[\![\text{not}]\!]\,(v) : b} \multimap E}{\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b} \Diamond_p I}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b\ \boxed{\text{C}}} \Diamond_p E_4$$

$$\dfrac{\dfrac{\dfrac{[\![\text{Sandy}]\!] : \Diamond_p s_2 \qquad \boxed{\text{C}}}{[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b} \Diamond_p E_3 \qquad [w : \Diamond_p s_1]_5}{w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b} \Diamond_p E_2}{\eta_{ci}(w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v))) : \Diamond_{ci} \Diamond_p b\ \boxed{\text{D}}} \Diamond_{ci} I$$

$$\dfrac{\boxed{\text{A}} \qquad \boxed{\text{D}}}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v))) : \Diamond_{ci} \Diamond_p b}\ \Diamond_{ci} E_5$$

$\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\delta^p_{ci}(\lambda i.\,[\![\text{goddamn}]\!]\,([\![\text{Sandy}]\!]\,(i))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.\,[\![\text{Sandy}]\!] \star_p \lambda z.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(w(i))(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\eta_p([\![\text{is}]\!]\,(w(i))([\![\text{Sandy}]\!]\,(i)))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.believe(kim, [\![\text{is}]\!]\,(w(i))([\![\text{Sandy}]\!]\,(i))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.\,[\![\text{not}]\!]\,(believe(kim, [\![\text{is}]\!]\,(w(i))([\![\text{Sandy}]\!]\,(i)))))$

$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\langle \lambda i.\,[\![\text{not}]\!]\,(believe(kim, [\![\text{is}]\!]\,(w(i))([\![\text{Sandy}]\!]\,(i)))), \varepsilon\rangle$

$\langle \lambda i.\,[\![\text{not}]\!]\,(believe(kim, [\![\text{is}]\!]\,([\![\text{Sandy}]\!]\,(i))([\![\text{Sandy}]\!]\,(i)))), bad(sandy)\rangle$

$\langle \lambda i.\neg believe(kim, [\![\text{Sandy}]\!]\,(i) = [\![\text{Sandy}]\!]\,(i)), bad(sandy)\rangle$

If we apply the speaker perspective to the first term of the pair we get the following reading:

$$\langle \neg believe(kim, sandy = sandy), bad(sandy)\rangle$$

5.3.3 Satisfiable Reading 1

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\![\text{goddamn}]\!] : s_1 \multimap \Diamond_{ci} s_1 \qquad [x : s_1]_1}{[\![\text{goddamn}]\!]\,(x) : \Diamond_{ci} s_1} \multimap E}{\eta_p([\![\text{goddamn}]\!]\,(x)) : \Diamond_p \Diamond_{ci} s_1} \Diamond_p I \qquad [\![\text{Sandy}]\!] : \Diamond_p s_1}{[\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x)) : \Diamond_p \Diamond_{ci} s_1} \Diamond_p E_1}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) : \Diamond_{ci} \Diamond_p s_1\ \boxed{\text{A}}} swap^p_{ci}$$

$$\dfrac{\dfrac{\dfrac{[\![\text{is}]\!] : s_1 \multimap s_2 \multimap i \qquad [y : s_1]_2}{[\![\text{is}]\!]\,(y) : s_2 \multimap i} \multimap E \qquad [z : s_2]_3}{[\![\text{is}]\!]\,(y)(z) : i} \multimap E}{\eta_p([\![\text{is}]\!]\,(y)(z)) : \Diamond_p i} \Diamond_p I \qquad [\![\text{Sandy}]\!] : \Diamond_p s_2}{[\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z)) : \Diamond_p i\ \boxed{\text{B}}} \Diamond_p E_3$$

$$\dfrac{\dfrac{[\![\text{Kim}]\!] : k \qquad [\![\text{believe}]\!] : k \multimap \Diamond_p i \multimap \Diamond_p b}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!]) : \Diamond_p i \multimap \Diamond_p b} \multimap E \qquad \boxed{\text{B}}}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) : \Diamond_p b} \multimap E \qquad \dfrac{\dfrac{[\![\text{not}]\!] : b \multimap b \qquad [v : b]_4}{[\![\text{not}]\!]\,(v) : b} \multimap E}{\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b} \Diamond_p I}{[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b\ \boxed{\text{C}}} \Diamond_p E_4$$

$$\dfrac{\dfrac{\dfrac{\boxed{\text{C}} \qquad [w : \Diamond_p s_1]_5}{w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)) : \Diamond_p b} \Diamond_p E_2}{\eta_{ci}(w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v))) : \Diamond_{ci} \Diamond_p b\ \boxed{\text{D}}} \Diamond_{ci} I$$

$$\dfrac{\boxed{\text{A}} \qquad \boxed{\text{D}}}{\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v))) : \Diamond_{ci} \Diamond_p b} \Diamond_{ci} E_5$$

$\delta^p_{ci}([\![\text{Sandy}]\!] \star_p \lambda x.\eta_p([\![\text{goddamn}]\!]\,(x))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$
$\delta^p_{ci}(\lambda i.\,[\![\text{goddamn}]\!]\,([\![\text{Sandy}]\!]\,(i))) \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(w \star_p \lambda y.\,[\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(y)(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.([\![\text{believe}]\!]\,([\![\text{Kim}]\!])([\![\text{Sandy}]\!] \star_p \lambda z.\eta_p([\![\text{is}]\!]\,(w(i))(z))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))(i))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.([\![\text{believe}]\!]\,([\![\text{Kim}]\!])(\lambda j.\,[\![\text{is}]\!]\,(w(i))([\![\text{Sandy}]\!]\,(j))) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))(i))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.(\lambda k.believe(kim, w(i) = [\![\text{Sandy}]\!]\,(kim)) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))(i))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.(\lambda k.believe(kim, w(i) = impostor) \star_p \lambda v.\eta_p([\![\text{not}]\!]\,(v)))(i))$
$\langle \lambda i.\,[\![\text{Sandy}]\!]\,(i), bad(sandy)\rangle \star_{ci} \lambda w.\eta_{ci}(\lambda i.\neg believe(kim, w(i) = impostor))$
$\langle \lambda i.\neg believe(kim, [\![\text{Sandy}]\!]\,(i) = impostor), bad(sandy)\rangle$

In this case, if we apply the speaker's perspective to the first term of the pair we get the following reading:

$$\langle \neg believe(kim, sandy = impostor), bad(sandy)\rangle$$

5.3.4 Satisfiable Reading 2

$$\dfrac{\dfrac{\dfrac{[\text{kim}] : k \qquad [\text{believe}] : k \multimap \Diamond_p i \multimap \Diamond_p b}{[\text{believe}]\,([\text{kim}]) : \Diamond_p i \multimap \Diamond_p b}\multimap E \qquad [x : \Diamond_p i]_1}{[\text{believe}]\,([\text{kim}])(x) : \Diamond_p b}\multimap E \qquad \dfrac{\dfrac{[\text{not}] : b \multimap b \qquad [y : b]_2}{[\text{not}]\,(y) : b}\multimap E}{\eta_p([\text{not}]\,(y)) : \Diamond_p b}\Diamond_p I}{[\text{believe}]\,([\text{kim}])(x) \star_p \lambda y.\eta_p([\text{not}]\,(y)) : \Diamond_p b}\Diamond_p E_2}{\eta_{ci}([\text{believe}]\,([\text{kim}])(x) \star_p \lambda y.\eta_p([\text{not}]\,(y))) : \Diamond_{ci}\Diamond_p b\ \boxed{\text{A}}}\Diamond_{ci} I$$

$$\dfrac{\dfrac{\boxed{\text{A}} \qquad [z : \Diamond_{ci}\Diamond_p i]_3}{z \star_{ci} \lambda x.\eta_{ci}([\text{believe}]\,([\text{kim}])(x) \star_p \lambda y.\eta_p([\text{not}]\,(y)))}\Diamond_{ci} E_1}{\lambda z.z \star_{ci} \lambda x.\eta_{ci}([\text{believe}]\,([\text{kim}])(x) \star_p \lambda y.\eta_p([\text{not}]\,(y))) : \Diamond_{ci}\Diamond_p i \multimap \Diamond_{ci}\Diamond_p b\ \boxed{\text{B}}}\multimap I_3$$

$$\dfrac{[\text{Sandy}] : \Diamond_p s_1 \qquad \dfrac{\dfrac{[\text{goddamn}] : s_1 \multimap \Diamond_{ci} s_1 \qquad [w : s_1]_4}{[\text{goddamn}]\,(w) : \Diamond_{ci} s_1}\multimap E}{\eta_p([\text{goddamn}]\,(w)) : \Diamond_p\Diamond_{ci} s_1}\Diamond_p I}{[\text{Sandy}] \star_p \lambda w.\eta_p([\text{goddamn}]\,(w)) : \Diamond_p\Diamond_{ci} s_1\ \boxed{\text{C}}}\Diamond_p E_4$$

$$\dfrac{\dfrac{\dfrac{[\text{is}] : s_1 \multimap s_2 \multimap i \qquad [v : s_1]_5}{[\text{is}]\,(v) : s_2 \multimap i}\multimap E \qquad [u : s_2]_6}{[\text{is}]\,(v)(u) : i}\multimap E}{\eta_{ci}([\text{is}]\,(v)(u)) : \Diamond_{ci} i}\Diamond_{ci} I \qquad [t : \Diamond_{ci} s_1]_7}{t \star_{ci} \lambda v.\eta_{ci}([\text{is}]\,(v)(u)) : \Diamond_{ci} i}\Diamond_{ci} E_5$$
$$\dfrac{}{\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\text{is}]\,(v)(u))) : \Diamond_p\Diamond_{ci} i\ \boxed{\text{D}}}\Diamond_p I$$

$$\dfrac{\dfrac{\dfrac{\boxed{\text{C}} \qquad \boxed{\text{D}}}{[\text{Sandy}] \star_p \lambda w.\eta_p([\text{goddamn}]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\text{is}]\,(v)(u))) : \Diamond_p\Diamond_{ci} i}\Diamond_p E_7}{\delta^p_{ci}([\text{Sandy}] \star_p \lambda w.\eta_p([\text{goddamn}]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\text{is}]\,(v)(u)))) : \Diamond_{ci}\Diamond_p i}\, swap^p_{ci} \qquad \boxed{\text{B}}}{\delta^p_{ci}([\text{Sandy}] \star_p \lambda w.\eta_p([\text{goddamn}]\,(w)) \star_p \lambda t.\eta_p(t \star_{ci} \lambda v.\eta_{ci}([\text{is}]\,(v)(u)))) \star_{ci} \lambda x.\eta_{ci}([\text{believe}]\,([\text{kim}])(x) \star_p \lambda y.\eta_p([\text{not}]\,(y))) : \Diamond_{ci}\Diamond_p b}\multimap E$$
$$\varphi : \Diamond_{ci}\Diamond_p b\ \boxed{\text{E}}$$

$$\dfrac{\dfrac{\dfrac{[\text{Sandy}] : \Diamond_p s_2}{\eta_{ci}([\text{Sandy}]) : \Diamond_{ci}\Diamond_p s_2}\Diamond_{ci} I}{\delta^{ci}_p(\eta_{ci}([\text{Sandy}])) : \Diamond_p\Diamond_{ci} s_2}\, swap^{ci}_p \qquad \dfrac{\dfrac{[r : \Diamond_{ci} s_2]_8 \qquad \boxed{\text{E}}}{r \star_{ci} \lambda u.\varphi : \Diamond_{ci}\Diamond_p b}\Diamond_{ci} E_6}{\delta^{ci}_p(r \star_{ci} \lambda u.\varphi) : \Diamond_p\Diamond_{ci} b}\, swap^{ci}_p}{\delta^{ci}_p(\eta_{ci}([\text{Sandy}])) \star_p \lambda r.\delta^{ci}_p(r \star_{ci} \lambda u.\varphi) : \Diamond_p\Diamond_{ci} b}\Diamond_p E_8$$
$$\dfrac{}{\delta^p_{ci}(\delta^{ci}_p(\eta_{ci}([\text{Sandy}])) \star_p \lambda r.\delta^{ci}_p(r \star_{ci} \lambda u.\varphi)) : \Diamond_{ci}\Diamond_p b}\, swap^p_{ci}$$

Doing all the reductions we get the following reading:

$$\langle \neg believe(kim, impostor = sandy), bad(sandy) \rangle$$

5.4 $\Diamond_p\Diamond_{ci}A$ and $\Diamond_{ci}\Diamond_pA$ are Almost Isomorphic

The last derivation poses the question regarding the order of the two monads in the final type, as the last step in the derivation is really just needed so that we have the same type for all four derivation. In this section we show that the order actually doesn't matter.

The composition of δ^p_{ci} and δ^{ci}_p (in both directions) generates something that looks like an isomorphism, but it's not a true isomorphism (it couldn't be). To see that we first define two natural transformations, pip between $\Diamond_p$ and the identity functor, and ai between $\Diamond_{ci}$ and again the identity functor. The first natural transformation "puts in perspective" a potentially contentious term by applying to it the default speaker perspective:

$$pip(x) = x(\sigma) \tag{5.11}$$

The second natural transformation extracts the at-issue material:

$$ai(x) = \pi_1(x) \tag{5.12}$$

The first thing we are going to prove is that their composition acts as a right identity for the composition of these two natural transformations:

$$A \xleftarrow{pip_A} \Diamond_pA \xleftarrow{ai_{\Diamond_pA}} \Diamond_{ci}\Diamond_pA \underset{\delta^p_{ci}}{\overset{\delta^{ci}_p}{\rightleftarrows}} \Diamond_p\Diamond_{ci}A \xrightarrow{pip_{\Diamond_{ci}A}} \Diamond_{ci}A \xrightarrow{ai_A} A \tag{5.13}$$

Proof. $pip_A \circ ai_{\Diamond_pA} \circ \delta^p_{ci} \circ \delta^{ci}_p = pip_A \circ ai_{\Diamond_pA}$:

$$\begin{aligned}
&pip_A(ai_{\Diamond_pA}(\delta^p_{ci}(\delta^{ci}_p(x)))) = \\
&pip_A(ai_{\Diamond_pA}(\delta^p_{ci}(\lambda i.\langle(\pi_1(x))(i), \pi_2(x)\rangle))) = \\
&pip_A(ai_{\Diamond_pA}(\langle\lambda i.\pi_1(\langle(\pi_1(x))(i), \pi_2(x)\rangle), \pi_2(\langle(\pi_1(x))(\sigma), \pi_2(x)\rangle)\rangle = \\
&pip_A(ai_{\Diamond_pA}(\langle\lambda i.(\pi_1(x))(i), \pi_2(x)\rangle)) = \\
&pip_A(\pi_1(\langle\lambda i.(\pi_1(x))(i), \pi_2(x)\rangle)) = \\
&pip_A(\lambda i.(\pi_1(x))(i)) = \\
&(\pi_1(x))(\sigma)
\end{aligned}$$

$$\begin{aligned}
&pip_A(ai_{\Diamond_pA}(x)) = \\
&pip_A(\pi_1(x)) = \\
&(\pi_1(x))(\sigma)
\end{aligned}$$

□

Proof. $ai_A \circ pip_{\Diamond_{ci}A} \circ \delta^{ci}_p \circ \delta^p_{ci} = ai_A \circ pip_{\Diamond_{ci}A}$

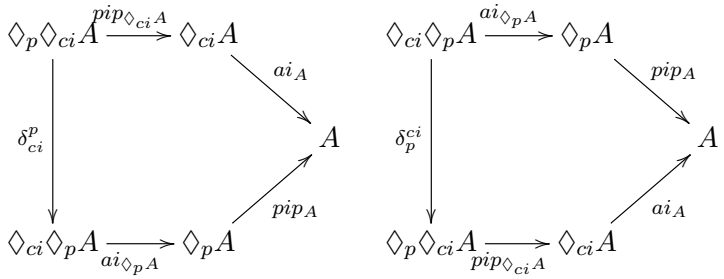
$$
\begin{aligned}
&ai_A(pip_{\Diamond_{ci}A}(\delta^{ci}_p(\delta^p_{ci}(x)))) = \\
&ai_A(pip_{\Diamond_{ci}A}(\delta^{ci}_p(\langle \lambda i.\pi_1(x(i)), \pi_2(x(\sigma))\rangle))) = \\
&ai_A(pip_{\Diamond_{ci}A}(\lambda i.\langle \pi_1(x(i)), \pi_2(x(\sigma))\rangle)) = \\
&ai_A(\langle \pi_1(x(\sigma)), \pi_2(x(\sigma))\rangle) = \\
&\pi_1(x(\sigma))
\end{aligned}
$$

$$
\begin{aligned}
&ai_A(pip_{\Diamond_{ci}A}(x)) = \\
&ai_A(x(\sigma)) = \\
&\pi_1(x(\sigma))
\end{aligned}
$$

□

This tells us that any iteration of an even number of "swapping" operations has no effect on the core meaning of the expression.

If we also prove that the following two diagrams commute, then we can extend the result to any number of iterations:



(5.14)

Proof. $pip_A \circ ai_{\Diamond_pA} \circ \delta^p_{ci} = ai_A \circ pip_{\Diamond_{ci}A}$

$$
\begin{aligned}
&pip_A(ai_{\Diamond_pA}(\delta^p_{ci}(x))) = \\
&pip_A(ai_{\Diamond_pA}(\langle \lambda i.\pi_1(x(i)), \pi_2(x(\sigma))\rangle)) = \\
&pip_A(\lambda i.\pi_1(x(i))) = \\
&\pi_1(x(\sigma))
\end{aligned}
$$

$$
\begin{aligned}
&ai_A(pip_{\Diamond_{ci}A}(x)) = \\
&ai_A(x(\sigma)) = \\
&\pi_1(x(\sigma))
\end{aligned}
$$

□

Proof. $ai_A \circ pip_{\Diamond_{ci}A} \circ \delta_p^{ci} = pip_A \circ ai_{\Diamond_pA}$

$$
\begin{aligned}
&ai_A(pip_{\Diamond_{ci}A}(\delta_p^{ci}(x))) = \\
&ai_A(pip_{\Diamond_{ci}A}(\lambda i.\langle \pi_1(x)(i), \pi_2(x)\rangle)) = \\
&ai_A(\langle \pi_1(x)(\sigma), \pi_2(x)\rangle) = \\
&\pi_1(x)(\sigma)
\end{aligned}
$$

$$
\begin{aligned}
&pip_A(ai_{\Diamond_pA}(x)) = \\
&pip_A(\pi_1(x)) = \\
&\pi_1(x)(\sigma)
\end{aligned}
$$

□

The second thing we want to prove is that the swapping operations don't alter the side-issue material. We define a polymorphic function $si : \forall a.\Diamond_{ci}a \to p$, where p is the type of the monoid of propositions:[1]

$$si(x) = \pi_2(x) \tag{5.15}$$

The function simply extracts the side-issue material from the monad.

$$
\begin{array}{ccc}
\Diamond_{ci}\Diamond_pA & \xrightarrow{si} & p \\
{\scriptstyle \delta_p^{ci}}\downarrow & & \uparrow{\scriptstyle si} \\
\Diamond_p\Diamond_{ci}A & \xrightarrow[pip_{\Diamond_{ci}A}]{} & \Diamond_{ci}A
\end{array}
\qquad
\begin{array}{ccc}
\Diamond_p\Diamond_{ci}A & \xrightarrow{pip_{\Diamond_{ci}A}} & \Diamond_{ci}A \\
{\scriptstyle \delta_{ci}^{p}}\downarrow & & \downarrow{\scriptstyle si} \\
\Diamond_{ci}\Diamond_pA & \xrightarrow[si]{} & m
\end{array}
\tag{5.16}
$$

Proof. $si \circ pip_{\Diamond_{ci}A} \circ \delta_p^{ci} = si$

$$
\begin{aligned}
&si(pip_{\Diamond_{ci}A}(\delta_p^{ci}(x))) = \\
&si(pip_{\Diamond_{ci}A}(\lambda i.\langle \pi_1(x)(i), \pi_2(x)\rangle)) = \\
&si(\langle \pi_1(x)(\sigma), \pi_2(x)\rangle) = \\
&\pi_2(x)
\end{aligned}
$$

$$si(x) = \pi_2(x)$$

□

[1] This thing is not a natural transformation, what is it?

Proof. $si \circ pip_{\Diamond_{ci}A} = si \circ \delta^p_{ci}$

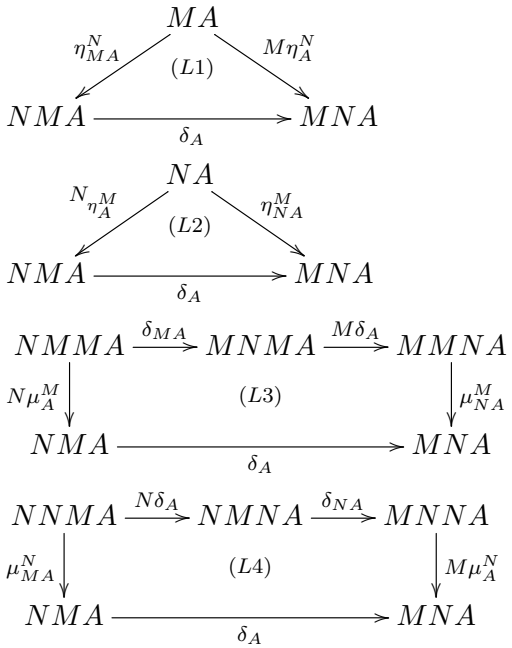
$$
\begin{aligned}
&si(pip_{\Diamond_{ci}A}(x)) = \\
&si(x(\sigma)) = \\
&\pi_2(x(\sigma))
\end{aligned}
$$

$$
\begin{aligned}
&si(\delta^p_{ci}(x)) = \\
&si(\langle \lambda i.\pi_1(x(i)), \pi_2(x(\sigma))\rangle = \\
&\pi_2(x(\sigma)) =
\end{aligned}
$$

□

5.5 Proof of Distributivity

A distributive law $\delta : NM \to MN$ is a natural transformation that must satisfy the following constraints (expressed as commuting diagrams):



Proposition 1. *δ^p_{ci} is a distributive law*

Proof. Definitions:

- $\delta^p_{ci}(f) = \langle \lambda i.\pi_1(f(i)), \pi_2(f(\sigma))\rangle$
- $\eta^p(x) = \lambda i.x$
- $map^p(f) = \lambda x.\lambda i.f(x(i))$

- $\mu^p(f) = \lambda i.f(i)(i)$
- $\eta^{ci}(x) = \langle x, \varepsilon \rangle$
- $map^{ci}(f) = \lambda\langle x, m\rangle.\langle f(x), m\rangle$
- $\mu^{ci}(x) = \langle \pi_1(\pi_1(x)), \pi_2(\pi_1(x)) \cdot \pi_2(x)\rangle$

δ^p_{ci} is a natural transformation: In the proof we use an equivalent definition for map^{ci}:

$$map^{ci}(f) = \lambda x.\langle f(\pi_1(x)), \pi_2(x)\rangle$$

$$\begin{aligned}
&map^{ci}(map^p(f))(\delta^p_{ci}(x)) = \\
&map^{ci}(map^p(f))(\langle \lambda i.\pi_1(x(i)), \pi_2(x(\sigma))\rangle) = \\
&\langle map^p(f)(\lambda i.\pi_1(x(i))), \pi_2(x(\sigma))\rangle = \\
&\langle \lambda i.f(\pi_1(x(i))), \pi_2(x(\sigma))\rangle =
\end{aligned}$$

$$\begin{aligned}
&\delta^p_{ci}(map^p(map^{ci}(f))(x)) = \\
&\delta^p_{ci}(\lambda i.map^{ci}(f)(x(i))) = \\
&\delta^p_{ci}(\lambda i.\langle f(\pi_1(x(i))), \pi_2(x(i))\rangle) = \\
&\langle \lambda i.\pi_1(\langle f(\pi_1(x(i))), \pi_2(x(i))\rangle), \pi_2(\langle f(\pi_1(x(\sigma))), \pi_2(x(\sigma))\rangle)\rangle = \\
&\langle \lambda i.f(\pi_1(x(i))), \pi_2(x(\sigma))\rangle
\end{aligned}$$

$L1$**:** $\delta^p_{ci}(\eta(\langle x, m\rangle)) = \delta^p_{ci}(\lambda i.\langle x, m\rangle) = \langle \lambda i.x, m\rangle = \langle \eta^p(x), m\rangle = map^{ci}(\eta^p)(\langle x, m\rangle)$

$L2$**:** $\delta^p_{ci}(map^p(\eta^{ci})(f)) = \delta^p_{ci}(\lambda i.\eta^{ci}(f(i)) = \delta^p_{ci}(\lambda i.\langle f(i), \varepsilon\rangle) = \langle \lambda i.\pi_1(\langle f(i), \varepsilon\rangle), \pi_2(\langle f(\sigma), \epsilon\rangle)\rangle = \langle \lambda i.f(i), \varepsilon\rangle = \langle f, \varepsilon\rangle = \eta^{ci}(f)$

$L3$**:**

$$\begin{aligned}
&\delta^p_{ci}(map^p(\mu^{ci})(f)) = \\
&\delta^p_{ci}(\lambda i.\mu^{ci}(f(i))) = \\
&\langle \lambda i.\pi_1(\mu^{ci}(f(i))), \pi_2(\mu^{ci}(f(\sigma)))\rangle = \\
&\langle \lambda i.\pi_1(\pi_1(f(i))), \pi_2(\pi_1(f(\sigma))) \cdot \pi_2(f(\sigma))\rangle =
\end{aligned}$$

$$\begin{aligned}
&\mu^{ci}(map^{ci}(\delta^p_{ci})(\delta^p_{ci}(f))) = \\
&\mu^{ci}(map^{ci}(\delta^p_{ci})(\langle \lambda i.\pi_1(f(i)), \pi_2(f(\sigma))\rangle)) = \\
&\mu^{ci}(\langle \delta^p_{ci}(\lambda i.\pi_1(f(i))), \pi_2(f(\sigma))\rangle) = \\
&\mu^{ci}(\langle\langle \lambda i.\pi_1(\pi_1(f(i))), \pi_2(\pi_1(f(\sigma)))\rangle, \pi_2(f(\sigma))\rangle) = \\
&\langle \lambda i.\pi_1(\pi_1(f(i))), \pi_2(\pi_1(f(\sigma))) \cdot \pi_2(f(\sigma))\rangle
\end{aligned}$$

$L4$**:**

$$\delta^p_{ci}(\mu^p(f)) =$$
$$\delta^p_{ci}(\lambda i.f(i)(i)) =$$
$$\langle \lambda i.\pi_1(f(i)(i)), \pi_2(f(\sigma)(\sigma))\rangle$$

$$map^{ci}(\mu^p)(\delta^p_{ci}(map^p(\delta^p_{ci})(f))) =$$
$$map^{ci}(\mu^p)(\delta^p_{ci}(\lambda j.\delta^p_{ci}(f(j)))) =$$
$$map^{ci}(\mu^p)(\delta^p_{ci}(\lambda j.\langle \lambda i.\pi_1(f(j)(i)), \pi_2(f(j)(\sigma))\rangle)) =$$
$$map^{ci}(\mu^p)(\langle \lambda k.\pi_1(\langle \lambda i.\pi_1(f(k)(i)), \pi_2(f(k)(\sigma))\rangle), \pi_2(\langle \lambda i.\pi_1(f(\sigma)(i)), \pi_2(f(\sigma)(\sigma))\rangle)\rangle) =$$
$$map^{ci}(\mu^p)(\langle \lambda k.\lambda i.\pi_1(f(k)(i)), \pi_2(f(\sigma)(\sigma))\rangle) =$$
$$\langle \mu^p(\lambda k.\lambda i.\pi_1(f(k)(i))), \pi_2(f(\sigma)(\sigma))\rangle =$$
$$\langle \lambda i.\pi_1(f(i)(i)), \pi_2(f(\sigma)(\sigma))\rangle$$

□

Missing readings: The sequent calculus, as implemented, doesn't generate all possible readings compatible with the monads. In particular, with the current swap rules, the instance of Sandy modified by the expressive is always interpreted outside of the scope of believe, while (as shown above) it should be possible to evaluate them inside the scope of believe. In the current situation we only get the readings in (5.17) and (5.18), while we don't get the reading in (5.19), where $\lambda x.x \star_{ci} \lambda y.\eta_{ci}(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Kim}\rrbracket)(y)$ is just an involved way to express $\Diamond^2_{ci}(\llbracket\text{believe}\rrbracket\,(\llbracket\text{kim}\rrbracket))$. There is also another reading that we should generate which corresponds to the case when the unmodified instance of Sandy is interpreted outside the scope of believe, while the modified one is interpreted in the scope of believe.

$$\begin{aligned}\delta^p_{ci}(\llbracket\text{Sandy}\rrbracket \star_p \lambda m.\delta^{ci}_p(\delta^p_{ci}(\llbracket\text{Sandy}\rrbracket \star_p \lambda k.\delta^{ci}_p(\llbracket\text{goddamn}\rrbracket\,(m)\star_{ci}\\ \lambda h.\eta_{ci}(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Kim}\rrbracket)(\eta_p(\llbracket\text{is}\rrbracket\,(h)(k))) \star_p \lambda z.\eta_p(\llbracket\text{not}\rrbracket\,(z)))))))\end{aligned} \tag{5.17}$$

$$\begin{aligned}\delta^p_{ci}(\llbracket\text{Sandy}\rrbracket \star_p \lambda m.\delta^{ci}_p(\llbracket\text{goddamn}\rrbracket\,(m) \star_{ci} \lambda h.\eta_{ci}(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Kim}\rrbracket)(\llbracket\text{Sandy}\rrbracket \star_p\\ \lambda k.\eta_p(\llbracket\text{is}\rrbracket\,(h)(k))) \star_p \lambda z.\eta_p(\llbracket\text{not}\rrbracket\,(z)))))\end{aligned} \tag{5.18}$$

$$\begin{aligned}(\lambda x.x \star_{ci} \lambda y.\eta_{ci}(\llbracket\text{believe}\rrbracket\,(\llbracket\text{Kim}\rrbracket)(y))(\delta^p_{ci}(\llbracket\text{Sandy}\rrbracket \star_p \lambda v.\,\llbracket\text{Sandy}\rrbracket \star_p\\ \lambda z.\eta_p(\llbracket\text{goddamn}\rrbracket\,(v) \star_{ci} \lambda w.\eta_{ci}(\llbracket\text{is}\rrbracket\,(w)(z)))))\end{aligned} \tag{5.19}$$

The good thing is that in all cases the CI component is always interpreted using the speaker perspective, as expected.

The cut rule is not admissible with swap!: example $\Diamond_p s, s \multimap \Diamond_{ci} s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_p \Diamond_{ci} f$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{s \vdash s \qquad \Diamond_{ci} s \vdash \Diamond_{ci} s}{s, s \multimap \Diamond_{ci} s \vdash \Diamond_{ci} s} \multimap L}{s, s \multimap \Diamond_{ci} s \vdash \Diamond_p \Diamond_{ci} s} \Diamond_p R}{\Diamond_p s, s \multimap \Diamond_{ci} s \vdash \Diamond_p \Diamond_{ci} s} \Diamond_p L}{\Diamond_p s, s \multimap \Diamond_{ci} s \vdash \Diamond_{ci} \Diamond_p s} swap^p_{ci} R \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Diamond_p s \vdash \Diamond_p s \qquad \Diamond_p f \vdash}{\Diamond_p s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_p f} \multimap L}{\Diamond_p s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_{ci} \Diamond_p f} \Diamond_{ci} R}{\Diamond_{ci} \Diamond_p s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_{ci} \Diamond_p f} \Diamond_{ci} L}{\Diamond_p s, s \multimap \Diamond_{ci} s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_{ci} \Diamond_p f} Cut}{\Diamond_p s, s \multimap \Diamond_{ci} s, \Diamond_p s \multimap \Diamond_p f \vdash \Diamond_p \Diamond_{ci} f} swap^{ci}_p R$$

Indeed, the swap rule doesn't respect the subformula property.

Bibliography

Maria Aloni. Individual concepts in modal predicate logic. *Journal of Philosophical Logic*, 34:1–64, 2005.

Scott AnderBois, Adrian Brasoveanu, and Robert Henderson. Crossing the appositive/at-issue meaning boundary. In Man Li and David Lutz, editors, *Proceedings of SALT 20*, pages 328–346, 2010.

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`.

Doug Arnold and Louisa Sadler. Pottsian LFG. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of LFG10*, pages 43–63. CSLI Publications, Stanford, CA, 2010.

Doug Arnold and Louisa Sadler. Resource splitting and reintegration with supplementals. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of LFG11*, pages 26–46. CSLI Publications, Stanford, CA, 2011.

Ash Asudeh. *Resumption as Resource Management*. PhD thesis, Stanford University, 2004.

Ash Asudeh. *The Logic of Pronominal Resumption*. Oxford University Press, Oxford, 2012.

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

John L. Austin. *How to Do Things with Words*. Oxford University Press, Oxford, 2 edition, 1975.

Steve Awodey. *Category Theory*. Oxford University Press, 2 edition, 2010.

Chris Barker and Chung-Chieh Shan. *Continuations and Natural Language*. Oxford University Press, Oxford, 2014.

Chris Barker, Raffaella Bernardi, and Chung chieh Shan. Principles of interdimensional meaning interaction. In Man Li and David Lutz, editors, *Proceedings of SALT 20*, pages 109–127, 2010.

Michael Barr and Charles Wells. *Category Theory for Computing Science*, volume 10. Prentice Hall, 1990.

David Beaver. Pragmatics (to a first approximation). In Jelle Gerbrandy, Maarten Marx, Maarten de Rijke, and Yde Venema, editors, *JFAK — Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday*. Amsterdam University Press, 1999.

Nick Benton, Gavin M. Bierman, and Valeria de Paiva. Computational types from a logical perspective. *Journal of Functional Programming*, 8(2):177–193, 1998.

David Braun and Jennifer Saul. Simple sentences, substition, and mistaken evaluations. *Philosophical Studies*, 111(1):1–41, 2002.

Tyler Burge. Individualism and the mental. *Midwest Studies in Philosophy*, 4(1):73–121, 1979.

Daniel Büring. *Binding Theory*. Cambridge University Press, Cambridge, 2005.

Bruce W Carlson and J Frank Yates. Disjunction errors in qualitative likelihood judgment. *Organizational Behavior and Human Decision Processes*, 44(3):368–379, 1989.

Héctor-Neri Castañeda. Thinking and the structure of the world: *Discours d'ontologie*. *Crítica: Revista Hispanoamericana de Filosofía*, 6(18):43–86, 1972.

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.

Gennaro Chierchia. Anaphora and attitudes *De Se*. In Renate Bartsch, Johan van Benthem, and Peter van Emde Boas, editors, *Language and Contextual Expressions*, pages 1–31. Foris, Dordrecht, 1989.

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. Language and nature. *Mind*, 104(416):1–59, 1995.

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

Maxwell J. Cresswell and Arnim von Stechow. *De Re* belief generalized. *Linguistics and Philosophy*, 5:503–535, 1982.

Robert Cummins. Methodological reflections on belief. In Radu J. Bogdan, editor, *Mind and Common Sense*, pages 53–70. Cambridge University Press: Philosophical Essays on Commonsense Psychology, Cambridge, 1991.

Haskell B. Curry and Robert Feys. *Combinatory Logic*, volume 1. North-Holland, Amsterdam, 1958.

Philippe de Groote, editor. *The Curry-Howard Isomorphism*, volume 8 of *Cahiers du Centre de Logique*. Academia, Louvain-la-neuve, Belgium, 1995.

Thomas Ernst. Speaker-oriented adverbs. *Natural Language and Linguistic Theory*, 27(3): 497–544, 2009.

Robert Fiengo and Robert May. Names and expressions. *Journal of Philosophy*, 95(8): 377–409, 1998.

Chris Fox and Shalom Lappin. *Foundations of Intensional Semantics*. Blackwell, Oxford, 2005.

Gottlob Frege. Function and concept. In Peter T. Geach and Max Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*, pages 22–41. Blackwell, Oxford, 1891/1952. Translation of *Funktion und Begriff*, in *Der Jenaischen Gesellschaft für Medizin und Naturwissenschaft*.

Gottlob Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

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.

Michele Giry. A categorical approach to probability theory. In *Categorical aspects of topology and analysis, Proc. int. Conf., Ottawa 1981, Lect. Notes Math. 915, 68-85 (1982).*, 1982.

Herbert Paul Grice. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics: Speech Acts*, volume 3, pages 41–58. Academic Press, San Diego, CA, 1975.

Daniel Gutzmann. *Use-Conditional Meaning. Studies in Multidimensional Semantics*. PhD thesis, University of Frankfurt, 2012.

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

Daniel Gutzmann and Hans-Martin Gärtner, editors. *Beyond Expressives: Explorations in Use-Conditional Meaning*. Brill, Leiden, 2013.

Ulrike Haas-Spohn. *Versteckte Indexikalität und subjektive Bedeutung*. Akademie Verlag, Berlin, 1995. English translation available as: *Hidden indexicality and subjective meaning*.

Claude Hagège. Les pronoms logophoriques. *Bulletin de la Société de Linguistique de Paris*, 69:287–310, 1974.

James A. Hampton. Overextension of conjunctive concepts: Evidence for a unitary model of concept typicality and class inclusion. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 14(1):12–32, 1988.

Irene Heim. Anaphora and semantic interpretation: A reinterpretation of Reinhart's approach. In Uli Sauerland and Orin Percus, editors, *The Interpretive Tract*, volume 25 of *MIT Working Papers in Linguistics*, pages 205–246. MITWPL, Cambridge, MA, 1998.

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.

Ralph Hertwig, Björn Benz, and Stefan Krauss. The conjunction fallacy and the many meanings of *and*. *Cognition*, 108:740–753, 2008.

Jaakko Hintikka. Semantics for propositional attitudes. In J.W̃. Davis, D.J̃. Hockney, and W.K̃. Wilson, editors, *Philosophical Logic*, pages 21–45. Reidel, Dordrecht, 1969.

Jaakko Hintikka. *The Intensions of Intensionality and Other New Models for Modalities*. Reidel, Dordrecht, 1975.

William A. Howard. The formulae-as-types notion of construction. In Jonathan P. Seldin and J. Roger Hindley, editors, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 479–490. Academic press, London, 1980. Circulated in unpublished form from 1969. Reprinted in de Groote [1995, 15–26].

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.

Theo M. V. Janssen. Compositionality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 495–553. Elsevier, London, 2 edition, 2011.

David Kaplan. Quantifying in. *Synthese*, 19(1-2):178–214, 1968. ISSN 0039-7857.

David Kaplan. Demonstratives. In Joseph Almog, John Perry, and Harvey Wettstein, editors, *Themes from Kaplan*, pages 481–563. Oxford University Press, Oxford, 1989.

Manfred Krifka. Quantifying into question acts. *Natural Language Semantics*, 9(1):1–40, 2001.

Manfred Krifka. Embedding illocutionary acts. In Thomas Roeper and Margaret Speas, editors, *Recursion: Complexity in Cognition*, pages 59–87. Springer International Publishing, 2014.

Saul Kripke. Naming and necessity. In Donald Davidson and Gilbert Harman, editors, *Semantics of Natural Language*, pages 253–355. Reidel, 1972.

Saul Kripke. A puzzle about belief. In Avishai Margalit, editor, *Meaning and Use*, pages 239–283. Reidel, Dordrecht, 1979.

Saul Kripke. *Naming and Necessity*. Harvard University Press, Cambridge, MA, 1980.

Alexander Kuhnle. Modeling uncertain data using monads and an application to the sequence alignment problem. Master's thesis, Karlsruhe Institute of Technology, 2013.

Susumo Kuno. *Functional Syntax: Anaphora, Discourse and Empathy*. University of Chicago Press, Chicago, 1987.

Richard Larson and Gabriel Segal. *Knowledge of Language: An Introduction to Semantic Theory*. MIT Press, Cambridge, MA, 1995.

Richard K. Larson and Peter Ludlow. Interpreted logical forms. *Synthese*, 95, 1993.

Peter Lasersohn. Context dependence, disagreement, and predicates of personal taste. *Linguistics and Philosophy*, 28(6):643–686, 2005.

Joe Lau and Max Deutsch. Externalism about mental content. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2014 edition, 2014.

Francis William Lawvere. The category of probabilistic mappings. Seminar handout, 1962.

David Lewis. Attitudes *De Dicto* and *De Se*. *Philosophical Review*, 88(4):513–543, 1979.

Peter Ludlow. Referential semantics for I-languages. In Louise M. Antony and Norbert Hornstein, editors, *Chomksy and His Critics*, pages 140–161. Blackwell, Oxford, 2003.

Hugh MacColl. Existential import of propositions. *Mind*, 14:401–402, 1905a.

Hugh MacColl. Symbolic reasoning. *Mind*, 14:490–397, 1905b.

John MacFarlane. *Assessment Sensitivity: Relative Truth and its Applications*. Clarendon Press, Oxford, 2014.

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

Barbara Mellers, Ralph Hertwig, and Daniel Kahneman. Do frequency representations eliminate conjunction effects? an exercise in adversarial collaboration. *Psychological Science*, 12:269–275, 2001.

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.

Richard Montague. The proper treatment of quantification in ordinary English. In Jaakko Hintikka, Julian Moravcsik, and Patrick Suppes, editors, *Approaches to Language*, pages 221–242. Reidel, Dordrecht, 1973. Reprinted in Montague [1974, 247–270].

Richard Montague. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, New Haven, CT, 1974. Edited and with an introduction by Richmond H. Thomason.

Michael Moortgat. Constants of grammatical reasoning. In Gosse Bouma, Erhard W. Hinrichs, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*, pages 199–219. CSLI Publications, Stanford, CA, 1999.

Michael Moortgat. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 95–179. Elsevier, second edition, 2011.

Rick Nouwen. *Plural pronominal anaphora in context*. PhD thesis, Utrecht Institute for Linguistics OTS, 2003.

David Y. Oshima. *Perspectives in Reported Discourse*. PhD thesis, Stanford University, Stanford, CA, 2006.

Barbara H. Partee. Lexical semantics and compositionality. In Lila R. Gleitman and Mark Liberman, editors, *An Invitation to Cognitive Science: Language*, volume 1, pages 311–360. MIT Press, Cambridge, MA, 2 edition, 1995.

Hazel Pearson. *The Sense of Self: Topics in the Semantics of* De Se *Expressions*. PhD thesis, Harvard University, 2013.

Orin Percus and Uli Sauerland. On the LFs of attitude reports. In *Sinn und Bedeutung*, volume 7, pages 228–242, 2003.

Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. The MIT Press, 1991.

David Pitt. Alter egos and their names. *The Journal of Philosophy*, 98(10):pp. 531–552, 2001. ISSN 0022362X.

David Pitt. Mental representation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2013 edition, 2013.

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

Christopher Potts. The expressive dimension. *Theoretical Linguistics*, 33(2):165–197, 2007.

Jim Pryor. Mental graphs. Ms., New York University, 2015.

Hilary Putnam. The meaning of 'meaning'. In *Mind, Language, and Reality*, volume II of *Philosophical Papers*, pages 215–271. Cambridge University Press, 1975.

Willard van Orman Quine. *From a Logical Point of View*. Harvard University Press, Cambridge, MA, 1953. Revised edition 1980.

Willard van Orman Quine. Quantifiers and propositional attitudes. *Journal of Philosophy*, 53:177–87, 1956.

Willard van Orman Quine. *Word and Object*. MIT Press, Cambridge, MA, 1960.

Bertrand Russell. On denoting. *Mind*, 14:479–493, 1905.

Nathan Salmon. *Frege's Puzzle*. MIT Press, Cambridge, MA, 1986.

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.

Stephen Schiffer. The mode-of-presentation problem. In C. Anthony Anderson and Joseph Owens, editors, *Propositional Attitudes: The Role of Content in Logic, Language and Mind*, pages 249–268. CSLI Publications, Stanford, CA, 1990.

Philippe Schlenker. A plea for monsters. *Linguistics and Philosophy*, 26(1):29–120, 2003.

Peter Sells. Aspects of logophoricity. *Linguistic Inquiry*, 18(3):445–479, 1987.

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.

Herbert A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956.

David I Spivak. Category theory for scientists (old version), 2013.

Tamina Stephenson. Judge dependence, epistemic modals, and predicates of personal taste. *Linguistics and Philosophy*, 30(4):487–525, 2007a.

Tamina Stephenson. *Towards a Theory of Subjective Meaning*. PhD thesis, MIT, 2007b.

Sandhya Sundaresan. *Context and (Co)rereference in the Syntax and its Interfaces*. PhD thesis, University of Tromsø and University of Stuttgart, 2012.

Katya Tentori, Nicolao Bonini, and Daniel Osherson. The conjunction fallacy: a misunderstanding about conjunction? *Cognitive Science: A Multidisciplinary Journal*, 28(3): 467–477, 2004.

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

Philip Wadler. The essence of functional programming. In *POPL '92: Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 1–14, New York, NY, USA, 1992b.

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.

J. Frank Yates and Bruce W. Carlson. Conjunction errors: Evidence for multiple judgment procedures, including "signed summation". *Organizational Behavior and Human Decision Processes*, 37:230–253, 1986.

Thomas Ede Zimmermann. Scopeless quantifiers and operators. *Journal of Philosophical Logic*, 22(5):545–561, October 1993.

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

Thomas Ede Zimmermann. Monotonicity in opaque verbs. *Linguistics and Philosophy*, 29 (6):715–761, 2006.