Interpretive Intentions

Ash Asudeh
(joint work with Gianluca Giorgolo)

LLI Lab, ICS, Carleton University
November 25, 2013

1 Introduction

- In this talk, we will look at some issues that have to do with reference and modality. The data we will consider is represented by the examples in table 1.

	Non-Modal	**Modal**
Same RE	(1) Dr. Octopus *punched Spider-Man* but he didn't *punch Spider-Man*	(2) Reza doesn't *believe Jesus* is *Jesus*
Different REs	(3) Mary Jane *loves Peter Parker* but she doesn't *love Spider-Man*	(4) Reza doesn't *believe Hesperus* is *Phosphorus*

Table 1: Motivating examples (where RE stands for "referring expression")

- Along the horizontal axis, expressions vary for the presence or absence of modal markers: either we have a purely assertive context (the here and now of the utterance), or we have some linguistic marker that changes the interpretive perspective of the expression.
- The vertical axis represents different ways to refer to the same entity: either we use the same expression twice or we use a different but co-extensional expression the second time.[1]
- The interplay of these two seemingly unrelated parameters generates some challenging readings for the expressions we want to discuss.
- Examples such as (4) are quite familiar from the literature on intensionality and constitute one of Frege's puzzles, sometimes referred to in terms of opacity, since the complement of propositional attitude verbs like *believe* does not allow truth-preserving substitution of co-extensional terms.
- We will instead focus on the case of sentences like (2) and (3).
  - Example (2) is interesting because, in addition to the logically implausible reading that attributes to Reza the non-belief in a tautology (which is unsatisfiable in standard modal logics), we also have another reading in which Reza is said to not believe that the entity he calls Jesus has certain properties that the speaker believes the entity to have.[2]
  - Example (3) is interesting because standard treatments of intensionality, where *love* is not normally considered an intensional verb, would lead us to expect that the example is a contradiction, on a par with (1), but it is not.

[1] Two expressions are co-extensional iff they refer to the same individual in the model at the world of evaluation (the actual world unless otherwise stated).

[2] To sharpen this reading, imagine that the Reza in question is Reza Aslan, the author of *Zealot*, and that the speaker is Laura Green, Christian CNN host.

- We take a different approach than the standard one to examples like (3) and (4), which also allows us to generalize to examples like (2), while still deriving a contradiction for (1).
- Our account does not model the intensional context created by operators like *believe* in terms of modality (doxastic modality, in this case), but rather proposes a way to model agents' *intentions/perspectives* in a lexically controlled way that builds a modal perspective into the lexical entry for the verb. In short, we decouple intensionality from modality: they are compatible, but modality is not necessary for intensionality.
- Our approach allows us to lexically mark verbs as intentional/perspectival in a manner that crosscuts the standard intensional/extensional divide.
- Verbs such as *believe* and *love* are marked as perspectival (relative to the subject), whereas a verb like *punch* is not.
- In essence, we propose that certain verbs involve *mental attitudes* — which generalizes the notion of 'propositional attitude' classically associated with verbs like *believe* to verbs like *love*, which do not take propositional complements — while other words do not (e.g., verbs that predicate physical actions).
- This distinction can distinguish verbs that are otherwise close in meaning. For example, *kill* is not a mental attitude verb, whereas *murder* and *assassinate* are, because they involve an intention to kill.

2 Overview

3. The problem
4. The standard solution
5. Our approach
6. Background
7. Analysis
8. Conclusion

A. Appendix: proofs

3 The Problem

- Example (4) is an instance of opacity, one of Frege's puzzles.
- The puzzle is normally set up as follows. Even though *Hesperus is Phosphorous* is true (these are two different names for the planet Venus), it is nevertheless possible for the first of the following sentences to be true, while the latter is false.

(5) Reza believes Hesperus is Venus.

(6) Reza believes Phosphorus is Venus.

- Thus, the problem is normally characterized as the inability to substitute co-referential terms while preserving truth. Contexts that have this property, such as the complement of *believe* are often called opaque contexts.
- It is easy to see that example (4) is just a standard example of opacity.
- Example (2) is clearly another instance of opacity, but involves repetition of the same term, rather than different, co-extensional terms.
- Example (3), however, involves a predicate *love* which is not normally taken to be opaque in its complement. For example, if my neighbour is Joey Jo-Jo Shabadoo, then it is normally judged that "Mary loves Joey Jo-Jo Shabadoo" is true iff "Mary loves my neighbour" is true. However, we believe that this judgement fails to allow Mary's perspective or intentions to play a part: if she does not know that JJJS is my neighbour, she may concur with the former statement while denying the latter.

4 The Standard Solution

- The standard approach to opacity relies on a scopal account (e.g., Montague 1974a).
- There is an operator that takes an intensional complement (e.g., the verb *believe*) and the relevant nominal is treated as a generalized quantifier that can either scope outside or inside this operator.
- For example, sentence (7) would have the two readings in (8).

(7) Reza believes an alien arrived.

(8) a. $[\![\text{believe}]\!]([\![\text{reza}]\!], \exists x([\![\text{alien}]\!](x) \wedge [\![\text{arrive}]\!](x)))$

b. $\exists x([\![\text{alien}]\!](x) \wedge [\![\text{believe}]\!]([\![\text{reza}]\!], [\![\text{arrive}]\!](x)))$

- The reading in (8a) is standardly called the *de dicto* reading. The truth of this reading does not depend on there being an alien in the actual world, just in Reza's belief worlds.
- The reading in (8b) is traditionally called the *de re* reading. The truth of this reading depends on there being an alien in the actual world, but not necessarily in Reza's beliefs (maybe he doesn't believe that the individual in question is an alien).

4.1 Issues with the Standard Approach

1. The standard approach is to treat nominals as generalized quantifiers, which are scopal operators. This allows them to either take scope over or under other operators, including intensional operators. The problem is that this is too permissive: it overgenerates readings.

   - For example, sentence (2) has 4 operators of interest:[3] the two instances of *Jesus*, *believe*, and the negation.
   - This yields six non-equivalent readings:
     (a) $\textbf{Jesus}_1 > \textbf{not} > \textbf{believe} > \textbf{Jesus}_2$
     (b) $\textbf{Jesus}_1 > \textbf{Jesus}_2 > \textbf{not} > \textbf{believe}$
     (c) $\textbf{not} > \textbf{believe} > \textbf{Jesus}_1 > \textbf{Jesus}_2$
     (d) $\textbf{Jesus}_1 > \textbf{not} > \textbf{Jesus}_2 > \textbf{believe}$
     (e) $\textbf{not} > \textbf{Jesus}_1 > \textbf{Jesus}_2 > \textbf{believe}$
     (f) $\textbf{not} > \textbf{Jesus}_1 > \textbf{believe} > \textbf{Jesus}_2$
   - The first three readings are legitimate — we return to these below.
   - The last three readings should not be generated. This is more obvious if we consider a simpler example with an existential quantifier, rather than a proper name:

     (9) Reza does not believe some alien landed.

   - The scoping **not** > **some alien** > **believe**, which corresponds to #1e/#1f above, is logically equivalent to **every alien** > **believe** > **not** but this interpretation[4] is unavailable for example (9): there should not be a de re universal reading.

2. There is another way in which the scopal account overgenerates. It generates spurious structural ambiguities. For example, for each of the six readings above, there is an alternative, truth conditionally equivalent reading in which $\textbf{Jesus}_1$ is swapped with $\textbf{Jesus}_2$. This constitutes a computational problem, since many unnecessary derivations are generated.

3. If proper names are treated as rigid designators (Kripke, 1972, 1980), the scopal account has a problem with examples like (2). If *Jesus* refers to the same individual in all possible worlds, only the contradictory reading, in which Reza does not believe a tautology, is derived.

4. For examples like (2), the modal/scopal account has a problem if Reza coincidentally believes what the speaker believes. Suppose that Reza is Jewish and it turns out that Jesus was not the Messiah and the speaker also believes this. Then the de re reading in fact also results in a contradiction, but it should intuitively just be false.

5. The intensional account does not extend to examples like (3) (the Mary Jane example), because *love* is not plausibly an intensional verb. If *love* were treated as an intensional verb, then there should be a de dicto reading available for a sentence like the following:

   (10) Mary Jane loves a unicorn.

   However, no such reading is available: if the sentence is true, it entails that there is a unicorn; that is, it does not admit a case in which MJ is simply mistaken about the identity of the entity she loves (e.g., it is a donkey with a novelty unicorn horn strapped to its head).

[3] In fact, the subject *Reza* also denotes a generalized quantifier, which means that there are even more readings at play, but we set these aside.

[4] The interpretation is **Every (actual) alien is such that Reza believes they did not land**.

5 Our Approach

5.1 The Intuition and an Informal Overview of its Empirical Consequences

- The intuition behind our analysis is that what is required to make sense of the pattern of readings in (1)-(4) is a notion of a *relativized interpretation*, where the possibility of relativization is a lexical feature of predicates. Interpretations are modelled as functions from indices to values. For example, the interpretation of a proper name like *Peter Parker* is a function from indices to individuals.

- The speaker's interpretation is always available (effectively a default). However, predicates that predicate a mental attitude of one of their arguments additionally bring that argument's interpretation function into play.

- Let us see how this informally explains the data in table 1.

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

  – This example is a contradiction. We correctly derive this because *punch* is not a mental attitude predicate, so the only interpretation in play is the speaker's. This means that *Spider-Man* refers consistently to whatever individual the speaker's interpretation function maps the name to. Thus, (1) is tantamount to the claim that Dr. Octopus both did and did not punch a certain individual who the speaker calls 'Spider-Man', which is a contradiction.[5]

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

  – The verb *believe* does involve a mental attitude on the part of its subject, so there are two interpretations at play here: Reza's and the speaker's. Supposing that Reza and the speaker have different interpretations for the name *Jesus* (which is the only way to derive the interesting, non-contradictory reading), then we derive three readings.

    1. For some entity x who Reza calls Jesus, Reza does not believe that x is x.
       [Contradiction]
    2. For some entity y who the speaker calls Jesus, Reza does not believe that y is y.
       [Contradiction]
    3. For some entity x who Reza calls Jesus and some entity y who the speaker calls Jesus, Reza does not believe that x is y.
       [Non-contradiction]

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

  – Suppose that the speaker knows that Peter Parker is Spider-Man, but MJ does not. In this circumstance, (3) is not a contradiction. We derive this as follows, using MJ's interpretation, assuming that *love* involves a mental attitude on the part of its subject.

    1. For some entity x who MJ calls Peter Parker and some entity y who MJ calls Spider-Man, such that $x \neq y$ for MJ, Mary Jane loves x, but Mary Jane does not love y.

[5] Note that we set aside the case where the second 'Spider-Man' is given contrastive focus.

(4) Reza doesn't believe Hesperus is Phosphorus.

- This is a classic case of Fregean opacity. There are a number of non-contradictory readings possible, which we derive:[6]

  1. For some entity x which Reza calls Hesperus and some entity y which Reza calls Phosphorus, such that $x \neq y$ for Reza, Reza does not believe that x is y.
  2. For some entity x which the speaker calls Hesperus and some entity y which the speaker calls Phosphorus, such that $x \neq y$ for Reza, Reza does not believe that x is y.
  3. For some entity x which Reza calls Hesperus and some entity y which the speaker calls Phosphorus, such that $x \neq y$ for Reza, Reza does not believe that x is y.
  4. For some entity x which the speaker calls Hesperus and some entity y which Reza calls Phosphorus, such that $x \neq y$ for Reza, Reza does not believe that x is y.

5.2 How We Formalize the Intuition (Informally)

- Our analysis follows a recent suggestion in the formal semantics literature that tries to extend the expressive power of the meaning language in a principled and controlled way with the use of (strong) monads (Shan, 2001; Unger, 2011; Giorgolo and Asudeh, 2012a).[7]

- Monads are mathematical structures that have found application in the theory of programming language semantics to model 'impure' computations.

- They encapsulate the notion of mapping a space of values (for instance the values of a simply-typed theory constructed over the basic e and t types) to a space populated by more complex values.

- The fundamental property of monadic mappings is that they allow us to freely combine standard simple values with complex ones.

- This allows us to consider more complex types of meaning only when truly necessary, avoiding the notorious problem of generalizing our lexical entries to the worst case.[8]

6 Background

6.1 Monads

- The concept of a *monad* originates in category theory to describe a special kind of *endofunctor* (think: structure-preserving mapping) that have the algebraic properties of a *monoid* (think of an object of a certain kind that can be combined with another one to obtain the same kind of object, and such that there is a special object that when combined with others doesn't change them; e.g., natural numbers with + and 0.).

- For our purposes, we will think of monads as structure-preserving mappings from 1) a collection of types (named sets of objects) and mappings (functions) between these types to 2) another collection of related types and mappings that we consider more complex.

- Structure-preservation means for us that things like the result of applying a function to an argument, or the composition of two functions (think iterated application as in the case of a sentence level operator applied to a predicate and its argument) should be preserved in the more complex image (modulo the additional structure that comes with the new types).

[6] Note that relative to Reza's beliefs, the first reading is like the reading above for (3).

[7] Shan (2001) was the first to suggest to use monads to model intensionality. We follow his suggestion and extend it to other cases, but we use a different formalization derived from the one presented in Benton et al. (1998).

[8] For example, Montague (1974b) treats all of the non-logical vocabulary as intensional, which leads to the counterintuitive result that the analysis of what, for Montague, are simple, extensional predicates like *find* is more complex than the analysis of intensional predicates like *seek*.

  - For example, the operation of pairing any object with another element of a *fixed* type is a monad. The image types are simply the pairing of the original types with the fixed type, and the original functions are mapped so that they only operate on the values of the original type.

- There are different ways to define a monad. Here we use a method that is very common in computer science.

- A monad is defined by three things:

  1. The functor that associates the original types with the new types and associates functions between the original types with functions between the new types.
  2. An operation called *unit* that maps the actual values of the original types to their images under the functor.
  3. An operation called *bind* that functions similarly to functional application (although it is normally written down in reverse: argument first and function second) and that allows us to "extract" the original values from their new more complex state and use them to generate new monadic values.

- We write this triple as $\langle M, \eta, \star \rangle$, where M is the functor, η the unit and $\star$ the bind.

- The type of η is $\forall a.a \to M(a)$.

- The type of $\star$ is $\forall a.\forall b.M(a) \to (a \to M(b)) \to M(b)$. (So $\star$ is like a flipped around functional application that would have the type $(a \to b) \to a \to b$; we use this flipped version to save on parentheses when we write our terms.).

- There are three laws that guarantee that what we have is a proper monad:

$$\eta(x) \star f = f(x) \tag{11}$$
$$m \star \eta = m \tag{12}$$
$$(m \star f) \star g = m \star (\lambda x.f(x) \star g) \tag{13}$$

  - With (17) and (12) we are sure that η doesn't do anything special to our values or attaches any additional information to them. Basically we want η to be like 0 with respect to $+$, but in this case η brings us to a new type (although we also have the trivial identity monad).
  - (13) simply states that $\star$ must be associative. But notice that bind doesn't have to be commutative, although in some cases it is.[9]

[9] There are also other laws that a monad has to satisfy and that are related to its functorial nature. We skip them because we would need to discuss the categorical nature of monads.

- As an example, we define the monad that arises from the operation of parametrizing a value over some index of type i (it's not important what i is, you can replace it with whatever you want, for instance the type of worlds in a possible worlds semantics).

  – The functor — which in this case we call R for "reader", given that this is its usual name in computer science (as it models the operation of producing values that depend on some environment that is read) — maps every type τ to the type $i \to \tau$, i.e. the type of functions from indices to objects of τ. R maps every function $f : a \to b$ to a function $R(f) : R(a) \to R(b)$, or unwrapping the R on types $R(f) : (i \to a) \to (i \to b)$. Operationally $R(f)$ is just the composition of its first argument of type $i \to a$ with our old f:

  $$R(f) = \lambda x.f \circ x \tag{14}$$

  – The unit for this monad takes a value and returns a constant function that always returns that value independently of the index passed. In other words, unit makes sure that the original value is *not* actually parametrized (remember that it has to behave as a kind of 0). Here is how we define it:

  $$\eta(x) = \lambda i.x \tag{15}$$

  – Finally $\star$ basically work as a power version of functional application that also takes care of passing around the index passed as argument:

  $$m \star f = \lambda i.f(m(i))(i) \tag{16}$$

6.2 Monadic Glue

- We assume Glue Semantics (Dalrymple, 1999, 2001; Asudeh, 2012) as our compositional system, but with a monadic meaning language (Giorgolo and Asudeh, 2012a,b).

- Glue Semantics assumes linear logic as the logic that 'glues' meanings together, performing composition. The primary connectives for our analysis are linear implication, $\multimap$, and the modality, $\Diamond$ (following Benton et al. 1998). The natural deduction proof rules for these connectives are shown in figure 1.

$$\dfrac{\begin{matrix}[x : a]_i \\ \vdots \\ t : b\end{matrix}}{\lambda x.t : a \multimap b} \multimap I_i \qquad \dfrac{x : a \qquad f : a \multimap b}{f(x) : b} \multimap E$$

$$\dfrac{x : a}{\eta(x) : \Diamond a} \Diamond I \qquad \dfrac{m : \Diamond a \qquad \begin{matrix}[x : a]_i \\ \vdots \\ n : \Diamond b\end{matrix}}{m \star \lambda x.n : \Diamond b} \Diamond E_i$$

Figure 1: Key natural deduction rules

7 Analysis

- The monad we use formalizes the intuition that the interpretation of certain referring expressions varies according to the perspective taken.
- We represent the dependency on a specific point of view by parametrizing our values over interpretation indices (in our case, simply entities).
- Technically, our monad maps values of type τ to values of type $i \to \tau$, i.e. functions from indices to values of type τ.
- We will refer to such a lifted τ type with $\Diamond\tau$.
- The monad is then defined by the unit operation η, that tells us how values are lifted from the original value space to the monadic one, and the bind operator $\star$, a sort of functional application that combines two monadic values to generate a new one:

$$\eta(x) = \lambda i.x : \tau \to \Diamond\tau \tag{17}$$
$$m \star k = \lambda i.k(m(i))(i) : \Diamond\tau \to (\tau \to \Diamond\delta) \to \Diamond\delta \tag{18}$$

- Intuitively, the unit operation just wraps the simple value with a vacuous abstraction, without making it dependent on the interpretation index.
- The bind operation simply distributes the interpretation index between the two composed monads.
- We focus on the analysis of sentence (2), which is particularly challenging for standard accounts that assume an intensional/possible worlds approach, where *believe* is modelled as a doxastic modal and names are rigid designators.

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

- Our analysis is based on the lexicon shown in table 2.
- Names that have possibly more than one interpretation are assigned the type $\Diamond e$.[10] The denotation of $\Diamond e$ is a function that returns a different entity according to the interpretation index that is the input (σ is the default index associated with the speaker/listener interpretation).
- We are agnostic regarding the nature of entities, as our analysis does not depend on an ontological specification of entities.
- The modal *believe* replaces the default index for the interpretation of its complement by replacing it with its subject.
- We thus obtain a mixture of modal models between the speaker/listener model and the subject's.

Word	Denotation	Type	Word	Denotation	Type
Reza	$\mathbf{r}$	e	not	$\lambda p.\neg p$	$t \to t$
Jesus	$\lambda i.\begin{cases}\mathbf{j_1} & \text{if } i = \mathbf{r},\\ \mathbf{j_2} & \text{if } i = \sigma\end{cases}$	$\Diamond e$	is	$\lambda x.\lambda y.x = y$	$e \to e \to t$
			believe	$\lambda s.\lambda c.\mathbf{B}(s, c(s))$	$e \to \Diamond t \to \Diamond t$

Table 2: Lexicon

- We obtain readings in which the instances of the name *Jesus* are interpreted outside of the scope of the doxastic operator because the monads to which they correspond can be evaluated before the evaluation of the monad associated with the operator.

[10] This should be the type for all names, but for simplicity of exposition, we assign only *Jesus* this type.

- We thus obtain a second form of scoping, independent from the one associated with functional application, which therefore prevents unwanted interactions with other operators (e.g., negation).
- Our analysis results in the following three readings:[11]

(19) $[\![\text{believe}]\!]\,(\eta([\![\text{Reza}]\!]))([\![\text{Jesus}]\!] \star \lambda x.\,[\![\text{Jesus}]\!] \star \lambda y.\eta([\![\text{is}]\!]\,(x)(y))) \star \lambda t.\eta([\![\text{not}]\!]\,(t))$

(20) $[\![\text{Jesus}]\!] \star \lambda x.\,[\![\text{Jesus}]\!] \star \lambda y.\,[\![\text{believe}]\!]\,(\eta([\![\text{Reza}]\!]))(\eta([\![\text{is}]\!]\,(x)(y))) \star \lambda t.\eta([\![\text{not}]\!]\,(t))$

(21) $[\![\text{Jesus}]\!] \star \lambda x.\,[\![\text{believe}]\!]\,(\eta([\![\text{Reza}]\!]))([\![\text{Jesus}]\!] \star \lambda y.\eta([\![\text{is}]\!]\,(x)(y))) \star \lambda t.\eta([\![\text{not}]\!]\,(t))$

- (19) and (20) are two instances of the non-plausible (*de dicto*) reading. The first represent the reading that Reza does not believe that the entity he calls Jesus is the entity he calls Jesus. The second represents the reading that Reza does not believe that the entity that the speaker calls Jesus is the entity that the speaker calls Jesus.
- (21) corresponds to the satisfiable reading that Reza does not believe that the entity the speaker calls Jesus is the same entity that Reza calls Jesus.
- These three readings cover precisely the space of possible interpretation for (2).
- Proofs for the three readings are provided in the appendix.
- It is not difficult to see that this approach can be easily extended to the more traditional intensional cases. In the case of (4) we get the usual reading by letting *Hesperus* and *Phosphorus* be interpreted according to the speaker's and Reza's perspective (or *vice versa*).
- We also cover the non-intensional mental attitude cases like (3): we let the object of the verb *love* be interpreted according to the subject's perspective. Even though the speaker may know that Peter Parker and Spider-Man are the same entity, this is not necessarily the case for Mary Jane.

8 Conclusion

- The monadic approach allows a more structured meaning language in which we model perspectives/mental attitudes in terms of interpretation functions.
- We do not need to assume a possible worlds semantics (although our approach is not incompatible with one) and do not need to assume an account of de re/de dicto readings based on scope ambiguity.
- Nevertheless, we can treat names as rigid designators, i.e. as referring consistently across contexts (for a given agent).
- Our results are general, accounting not just for standard cases of opacity due to different names/descriptions, but also for cases of opacity created by a single differentially interpreted name and for related non-intensional cases with mental attitude verbs like *love*.
- Indeed, our approach begins to give some shape to the notion of mental attitude in terms of perspectives modelled as interpretive intentions.

[11] The semantic derivation produces six readings, reduced to four because our monad is commutative, and further to three because we have the same referring expression repeated twice.

References

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

Benton, Nick, G. M. Bierman, and Valeria de Paiva. 1998. Computational Types from a Logical Perspective. *Journal of Functional Programming* 8(2): 177–193.

Dalrymple, Mary, ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Giorgolo, Gianluca, and Ash Asudeh. 2012a. $\langle M, \eta, \star \rangle$ Monads for Conventional Implicatures. In Ana Aguilar Guevara, Anna Chernilovskaya, and Rick Nouwen, eds., *Proceedings of Sinn und Bedeutung 16*, vol. 1, 265–278. MIT Working Papers in Linguistics.

—. 2012b. Missing Resources in a Resource-Sensitive Semantics. In Miriam Butt and Tracy Holloway King, eds., *Proceedings of the LFG12 Conference*, 219–239. Stanford, CA: CSLI Publications.

Kripke, Saul. 1972. Naming and Necessity. In D. Davidson and G. Harman, eds., *Semantics of Natural Language*, 253–355. Reidel.

—. 1980. *Naming and Necessity*. Cambridge, MA: Harvard University Press.

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

—. 1974b. The Proper Treatment of Quantification in Ordinary English. In Montague 1974a. Edited and with an introduction by Richmond H. Thomason.

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

Unger, Christina. 2011. Dynamic Semantics as Monadic Computation. In Manabu Okumura, Daisuke Bekki, and Ken Satoh, eds., *New Frontiers in Artificial Intelligence - JSAI-isAI 2011*, 68–81.

A Proofs for *Reza doesn't believe Jesus is Jesus*

$$
\dfrac{
\dfrac{\llbracket\text{Reza}\rrbracket : \Diamond r \quad \llbracket\text{believe}\rrbracket : \Diamond r \multimap \Diamond i \multimap \Diamond b}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket) : \Diamond i \multimap \Diamond b}\multimap E
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\llbracket\text{is}\rrbracket : j \multimap j \multimap i \quad [x : j]_1}{\llbracket\text{is}\rrbracket(x) : j \multimap i}\multimap E
\quad [y : j]_2}{\llbracket\text{is}\rrbracket(x)(y) : i}\multimap E
}{\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond I
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond E_2
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond E_1
}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\multimap E
\quad
\dfrac{\dfrac{\llbracket\text{not}\rrbracket : b \multimap b \quad [t : b]_3}{\llbracket\text{not}\rrbracket(t) : b}\multimap E}{\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond I
}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y))) \star \lambda t.\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond E_3
$$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\llbracket\text{Reza}\rrbracket : \Diamond r \quad \llbracket\text{believe}\rrbracket : \Diamond r \multimap \Diamond i \multimap \Diamond b}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket) : \Diamond i \multimap \Diamond b}\multimap E
\quad
\dfrac{
\dfrac{
\dfrac{\llbracket\text{is}\rrbracket : j \multimap j \multimap i \quad [x : j]_1}{\llbracket\text{is}\rrbracket(x) : j \multimap i}\multimap E
\quad [y : j]_2}{\llbracket\text{is}\rrbracket(x)(y) : i}\multimap E
}{\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond I
}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\multimap E
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda y.\, \llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\Diamond E_2
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{Jesus}\rrbracket \star \lambda y.\, \llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\Diamond E_1
\quad
\dfrac{\dfrac{\llbracket\text{not}\rrbracket : b \multimap b \quad [t : b]_3}{\llbracket\text{not}\rrbracket(t) : b}\multimap E}{\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond I
}{\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{Jesus}\rrbracket \star \lambda y.\, \llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\eta(\llbracket\text{is}\rrbracket(x)(y))) \star \lambda t.\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond E_3
$$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\llbracket\text{Reza}\rrbracket : \Diamond r \quad \llbracket\text{believe}\rrbracket : \Diamond r \multimap \Diamond i \multimap \Diamond b}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket) : \Diamond i \multimap \Diamond b}\multimap E
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\llbracket\text{is}\rrbracket : j \multimap j \multimap i \quad [x : j]_1}{\llbracket\text{is}\rrbracket(x) : j \multimap i}\multimap E
\quad [y : j]_2}{\llbracket\text{is}\rrbracket(x)(y) : i}\multimap E
}{\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond I
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y)) : \Diamond i}\Diamond E_2
}{\llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\multimap E
\quad \llbracket\text{Jesus}\rrbracket : \Diamond j}{\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y))) : \Diamond b}\Diamond E_1
\quad
\dfrac{\dfrac{\llbracket\text{not}\rrbracket : b \multimap b \quad [t : b]_3}{\llbracket\text{not}\rrbracket(t) : b}\multimap E}{\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond I
}{\llbracket\text{Jesus}\rrbracket \star \lambda x.\, \llbracket\text{believe}\rrbracket(\llbracket\text{Reza}\rrbracket)(\llbracket\text{Jesus}\rrbracket \star \lambda y.\eta(\llbracket\text{is}\rrbracket(x)(y))) \star \lambda t.\eta(\llbracket\text{not}\rrbracket(t)) : \Diamond b}\Diamond E_3
$$