Fuck compositionality

Patrick D. Elliott

March 7, 2019 – Emo2019

ZAS https://patrickdelliott.com

Introduction

Preliminaries

- *The goal*: to argue for a particular way of integrating *expressive content* into our existing compositional regime, using an extension of Charlow's (2014) monadic grammar.
- I'll claim that expressive adjectives such as fucking *take scope* – they give rise to *non-local readings* subject to syntactic restrictions, just like other scope-takers. I'll model this in the context of a pervasively continuized grammar.
- I'll ultimately suggest that once we integrate expressive content – via the **Writer** monad – into a grammar with the resources to handle indeterminacy, scope, and binding, the distinctive way in which expressive adjectives interact familiar logical operators falls out naturally.

The general framework: Meaning as computation

- Charlow's framework is one according to which semantic composition is *computation* – modelled as function-argument application. See also: Shan (2002b), Elliott (2019).
- Functional programmers have developed sophisticated techniques for integrating those aspects of programs that seem non-compositional in a purely compositional way – e.g., state sensitivity, indeterminacy.
- *Monads* are one such method. In the following, we'll be using monads to model various aspects of meaning. They aren't anything scary – just a type constructor and two type-shifters which together obey a particular set of laws.

Roadmap

- Expressive adjectives, very briefly.
- Using **Writer** to model expressive content.
- Non-local readings of expressive adjectives.
- Using continuations to model the scope of expressive adjectives.
- Indeterminacy, indefinites, and expressivity.
- Quantification and expressivity.

(1) "Fucking Ollie!? He's a fucking knitted scarf that twat. He's a fucking balaclava." *The Thick of It*, BBC

(2) "You shitting idiot." *Touching the Void*, David Greig

(3)

Die	*wollen*	*eine*	*verfickte*	*unterbezahlte*	*Putzfrau*	*einstellen,*
They	want	a	fucking	underpaid	cleaning-lady	hire,

nur	*weil*	*sie*	*"keine Zeit"*	*zum*	*Putzen*	*haben.*
only	because	they	"no time"	to	clean	have.

"They want to hire a fucking unpaid cleaning lady, just because they have "no time" to clean" *twitter*

- At a broad level of abstraction (see McCready 2012 for important exceptions), *Expressive Adjectives* (EAs) convey a *negative expressive attitude* towards some entity, be it an individual, a *kind* or something like a *state of affairs*.

(4) The fucking cat is being affectionate for once. ☹ $\iota x[\text{cat } x]$

- It's important to remember that ☹ is just a placeholder for a fully-fledged semantics; How exactly to cash out ☹ is an interesting and important question, but *not* one I'll be concerned with here.
- Rather, I'm going to be concerned with how expressions which contribute *expressive side-effects* interact with other aspects of our compositional regime.

The Writer monad

Writer is often used to log data in tandem with performing ordinary computations.

An example of where we might want to use **Writer** is, e.g., to define a fancy version of addition that writes to a log (represented as a string) whether the result is even or odd.

- $2 + 2$
  Result: 4 ⇝ "The result is even."
- $4 + 1$
  Result: 5 ⇝ "The result is even. The result is odd."

How does **Writer** accomplish this?

It enriches values with an *additional dimension* (sound familiar?) – let's call it the *log* for now. Here is the writer type-constructor:

(5) $\mathrm{W_b\ a} := \mathrm{(a, b)}$

Now we can define fancy addition in terms of **Writer**. It's of type (Int, String) → (Int, String) → (Int, String).

$$(+) := \lambda(n, s)\ .\ \lambda(m, s')\ .\ \left(n + m, s +\!\!+ s' +\!\!+ \begin{array}{l} \text{if even } n + m \\ \text{then “the result is even”} \\ \text{else “the result is odd”} \end{array} \right)$$

Writer iii

So, $(+)$ takes two fancy integers of type $\mathsf{W}_{\mathsf{String}}$ Int, and returns another fancy integer.

This begs the question, what if the two integers we want to add are non-fancy, i.e., there is no existing log information? We need to define a way of taking a normal integer, and turning it into a *trivial* fancy integer – i.e., one with an empty log. We call such a function from an ordinary value to a trivial fancy value *return*:

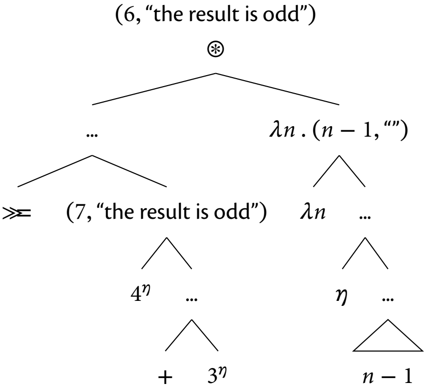
(6) Return: $n^{\eta} \coloneqq (n, \text{``''})$

We can also lift functions over normal integers into the fancy dimension with η:

(7) $(\lambda n \,.\, n - 1)^{\eta} = (\lambda n \,.\, n - 1, \text{``''})$

Finally, we can define a function to perform a computation while sequencing fancy side-effects. We call such a function *bind*.

(8) Bind: $(n, l) \ggcurly k \coloneqq (f\,n, l \mathbin{+\!\!+} l')$ where $(f, l') \coloneqq k\,n$



- **Writer**, as characterized, is a triple consisting of a type-constructor, a unary operation which we call *return*, and a binary operation which we call *bind*: $\langle M, \eta, \gg\!= \rangle$. Together, just so long as they obey certain laws, this constitutes a *monad*.
- I won't go through the monad laws here – they ensure that monads are well-behaved in certain important ways.
- In the context of linguistic semantics, we can think of *Monads* and related concepts as *theories of possible type shifters*.

An aside on monoids

- **Writer** comes with the requirement that b in W_b has a Monoid instance.
- Similarly to a monad, monoids are triples consisting of a type-constructor, as well as a value and an operation – $\varnothing$ and $\mathbin{+\!\!+}$.
- Like monads, monoids must obey two laws, which are simple enough to introduce here:

$$\overbrace{p \mathbin{+\!\!+} \varnothing = p}^{\text{identity}} \quad \overbrace{(p \mathbin{+\!\!+} q) \mathbin{+\!\!+} r = p \mathbin{+\!\!+} (q \mathbin{+\!\!+} r)}^{\text{associativity}}$$

- It is important that the log type has a monoid instance otherwise there is no way of glomping together log values without potentially losing information; $\varnothing$ provides the log for a trivially fancy value.

Expressive Writer

- Recall that we described the **Writer** monad informally as way of adding an *additional dimension of meaning*.
- In the literature on expressives, a common strategy has been to adopt a *multi-dimensional* semantics (see, e.g., McCready 2010).
- Building on Giorgolo & Asudeh (2012), Charlow (2015), we're going to take this parallel and run with it.
- We're going to model expressive content as a separate dimension via **Writer** – as we'll see, the projective nature and non-interaction of expressive content just falls out from this perspective.

Multi-dimensionality via Writer ii

Aping Potts, we use $(\bullet)$ to separate ordinary semantic values from expressive content – formally, this is just sugar for a pair constructor. t is a stand-in for your favourite propositional type.

(9) $\mathsf{W\ a} \coloneqq \mathsf{a} \bullet \mathsf{t}$

(10) Return: $x^{\eta} \coloneqq x \bullet \top$ $\qquad \mathsf{a} \to \mathsf{W\ a}$

(11) Bind: $(x \bullet p) \gg\!= k \coloneqq y \bullet p \wedge q$

where $y \bullet q \coloneqq k\,x$ $\qquad \mathsf{W\ a} \to (\mathsf{a} \to \mathsf{W\ b}) \to \mathsf{W\ b}$

We'll also provide an Apply function for convenience – this can be defined in terms of Return and Sequence.

(12) Apply: $(x \bullet p) \circledast (y \bullet q) \coloneqq \mathsf{A}\, x\, y \bullet p \wedge q$

$$\mathsf{W\,a \rightarrow W\,(a \rightarrow b) \rightarrow W\,b}$$

Apply just composes a fancy function with its fancy argument, while sequencing the side-effects associated with each.

The propositional conjunction monoid

t makes a suitable “log” type for Writer, since $\langle t, \wedge, \top \rangle$ is a monoid.

$\top$ (the tautology) is the identity for the propositional conjunction monoid...

(13) $\top \wedge p = p$

...and propositional conjunction is associative...

(14) $(p \wedge q) \wedge r = p \wedge (q \wedge r)$

Trivially fancy values receive $\top$ as their expressive content.

Associativity ensures that the expressive content of distinct constituents can be glomped together in a way that results in no loss of information.

A salient reading of (15) conveys the speaker's negative attitude towards *Lou*.

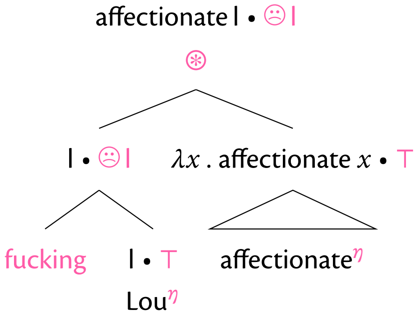
(15) Fucking Lou is being affectionate for once.

In order to account for this, we'll adopt a lexical entry for *fucking* which takes a fancy x, and bumps a negative attitude towards x into the expressive dimension.

(16) fucking $(x \bullet p) \coloneqq (x \bullet p \wedge ☹ x)$ $\qquad$ $\mathsf{W\,e} \to \mathsf{W\,e}$

A writer-monadic analysis of expressive adjectives ii

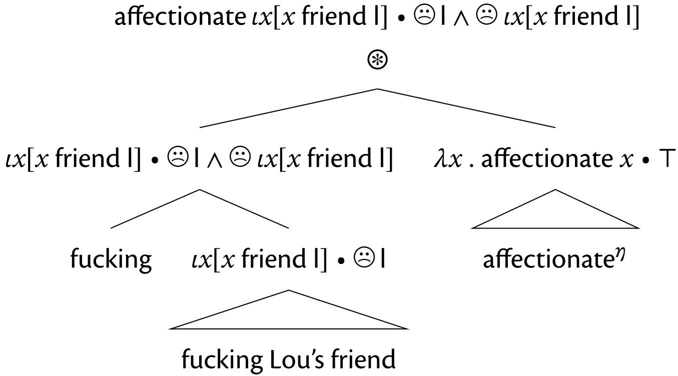
Composition proceeds via Writer's return (η) and apply ($\circledast$) operators.



Projectivity

(17) Fucking [fucking Lou's friend] is being affectionate for once.

(18)



We’ve been concentrating (and will continue to do so) on *functional expletive expressives* – those that contribute *only* expressive content, and no descriptive content, such as *damn* and *fucking* (Gutzmann 2013).

We’ll largely ignore *mixed expressives* such as pejoratives – those that contribute both descriptive and expressive content – but it’s straightforward to model them in a writer-monadic setting.

(19) $\text{mudblood} = (\lambda x\,.\,\text{muggle}\ x) \bullet \frownie\, \text{muggle}^{\cap}$

(20) $\text{mudblood} \circledast \text{Hermione}^{\eta} = \text{muggle h} \bullet \frownie\, \text{muggle}^{\cap}$

Capturing non-local readings

Gutzmann (2019) argues extensively that EAs give rise to so-called *non-local readings*. I will take his empirical claims to be essentially correct, aiming to answer the questions of *why* and *how*.

(21) The [fucking cat] is being affectionate for once. ☹ $\iota x[\text{cat } x]$

(21) can convey that the speaker has a negative attitude towards *the cat* – despite the fact that *fucking* takes as its sister just the NP *cat*. Note, importantly, that (21) is compatible with (i) the speaker having a positive attitude towards the situation, and (ii) the speaker having a positive attitude towards cats in general.

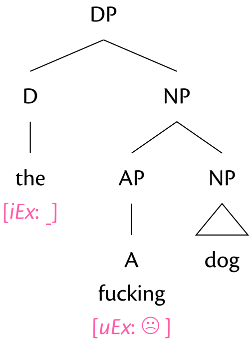
Both (22) and (23) can convey that the speaker has a negative attitude towards *the fact that the cat peed on the couch*: p. This, despite the fact that *fucking* is syntactically within the DP.

(22) The fucking cat (which I love)
is peeing on my favourite couch. ☹ p

(23) The cat is peeing on my favourite fucking couch. ☹ p

Gutzmann's syntactic account

Gutzmann (2019) claims that EAs come with an uninterpretable expressive feature, and the heads of constituents which can be the target of the expressive attitude come with an unvalued, interpretable expressive feature.



Via *upwards agree*, the uninterpretable expressive feature gets deleted,

Some straightforward worries

- Find me a language with some overt realization of expressive agreement!
- As we'll see later, the syntactic restrictions on non-local readings identified by Gutzmann pattern with *scope islands*.
- Nothing insightful to say about the interaction between expressive adjectives and quantificational determiners.

There are some pretty compelling reasons to conceptually disprefer an account based on expressive features that never have an overt morphological realization.

Instead, I'm going to pursue a scope-based account of non-local readings, which requires a short detour into continuation semantics...

Scope via Cont

The continuation monad **Cont** provides a fully compositional account of scope-taking – see Shan (2002a), Barker & Shan (2014), and Charlow (2014).

(24) $\mathsf{K_t\ a} \coloneqq (\mathsf{a} \to \mathsf{t}) \to \mathsf{t}$

(25) *Scopal lift:*
$x^{\uparrow} \coloneqq \lambda k \,.\, k\, x$ $\qquad \mathsf{a} \to \mathsf{K_t a}$

(26) *Scopal application*
$\mathsf{S}\ m\ n \coloneqq \lambda k \,.\, m\, (\lambda x \,.\, n\, (\lambda y \,.\, k\, (\mathsf{A}\ x\ y)))$

$$\begin{cases} \mathsf{K_t\ (a \to b) \to K_t\ a} \\ \mathsf{K_t\ a \to K_t\ (a \to b)} \end{cases} \to \mathsf{K\ b}$$

Tower notation

We can abbreviate *continuized* types and meanings using Barker & Shan's (2014) *tower notation*. The continuation type constructor encodes the result type on the top tier and the type of the contained ordinary value on the bottom tier.

(27) $\mathsf{K_t\ a} \coloneqq \dfrac{\mathsf{t}}{\mathsf{a}}$

Scopal lift takes a value and returns a trivial tower:

(28) $x^{\uparrow} \coloneqq \dfrac{[\,]}{x}$ $\qquad \mathsf{a} \rightarrow \dfrac{\mathsf{t}}{\mathsf{a}}$

Scopal application composes two towers – scopal side-effects get sequenced, and the contained, ordinary values undergo function application:

(29) $$\frac{f\,[]}{x}\;\mathsf{S}\;\frac{g\,[]}{y} := \frac{f\,[\,g\,[]]}{\mathsf{A}\,x\,y}$$

$$\frac{\mathsf{t}}{\mathsf{a} \rightarrow \mathsf{b}} \rightarrow \frac{\mathsf{t}}{\mathsf{a}} \rightarrow \frac{\mathsf{t}}{\mathsf{b}}$$

or

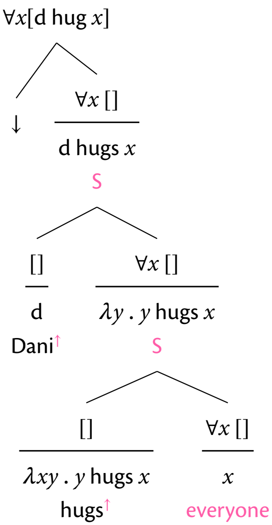
$$\frac{\mathsf{t}}{\mathsf{a}} \rightarrow \frac{\mathsf{t}}{\mathsf{a} \rightarrow \mathsf{b}} \rightarrow \frac{\mathsf{t}}{\mathsf{b}}$$

Finally, we need a way of lowering towers to ordinary values, in order to evaluate scopal side effects. We accomplish this by feeding a tower the identity function.

(30) $$\left(\frac{f[\,]}{p}\right)^{\downarrow} = f\,p \qquad \frac{\text{t}}{\text{t}} \rightarrow \text{t}$$

It's important to note, by the way, that towers are a *notational convenience* and have no independent representational status – unlike, e.g., Potts' parse trees – everything can be expressed in a purely model-theoretic (and therefore fully compositional) way.

The continuation monad and scope



Scope islands

A surprising fact – quantificational scope is roofed in certain syntactic environments – *scope islands*.

(31) Yasu crashed at least one bike [belonging to every linguist] (scope island) ✗ $\forall >$ at least one

(32) If [every ling prof is productive], the department will flourish. ✗ $\forall >$ if...then

(33) At least two profs reported [that every student passed]. ✗ $\forall >$ at least two

Scope islands ii

On the syntactic side, theories of scope islands haven't advanced much since May (1977). *Continuations* provide a unified *semantic* account of scope islands, grounded in the notion of a phase – the intuitive idea is that a scope island is a constituent at which no computations can be pending.

(34) *Definition:* (from Charlow 2014: p. 90)
A scope island is a constituent that must be evaluated.

We can characterize this formally in terms of *continuized types*, but the idea is intuitive – every continuation must be lowered via (↓) before a constituent is considered evaluated – no towers allowed.

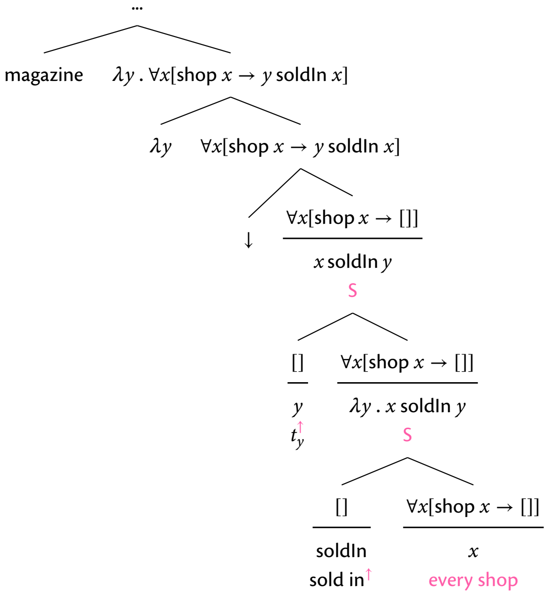
(35) Dani read at least one magazine sold in every shop. ✗ $\forall >$ at least one

(36) Unevaluated relative clause:

$$[\![\text{[sold in every shop]}]\!] = \frac{\forall x[\text{shop } x \to [\,]]}{\lambda y\,.\,y \text{ soldIn } x} \qquad \frac{\mathrm{t}}{\mathrm{e} \to \mathrm{t}}$$

If we try to lower (36) via (↓) we get a type mismatch, (the de-sugared type is $((\mathrm{e} \to \mathrm{t}) \to \mathrm{t}) \to \mathrm{t}$). The scope of the universal must be fixed pre-abstraction for the relative clause to be evaluated.

Scope islands iv



Consider again the type of a monad *M*'s sequencing operation (this time using tower notation to emphasise the correspondence):

(37) $(\gg\!=) \coloneqq \mathrm{M\,a} \to \dfrac{\mathrm{M\,b}}{\mathrm{a}}$ de-sugared: $\mathrm{M\,a} \to (\mathrm{a} \to \mathrm{M\,b}) \to \mathrm{M\,b}$

Essentially, $(\gg\!=)$ takes a fancy a, and returns a *monadic continuation*.

We'll use $(\gg\!=)$ to lift monadic values into trivial monadic scope takes. $(\gg\!=) \circ (\eta)$. Corresponds to monadic $(\uparrow)$. (η) corresponds to monadic $(\downarrow)$

(38) $(\uparrow) \coloneqq (\gg\!=) \circ (\eta)$

(39) $(\downarrow) \coloneqq (\eta)$

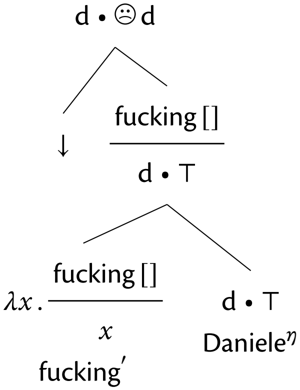
Expressive adjectives as scope-takers

Armed with our new compositional regime for dealing with scopal side-effects, we can lift our old meaning for **fucking** into a scope-taker:

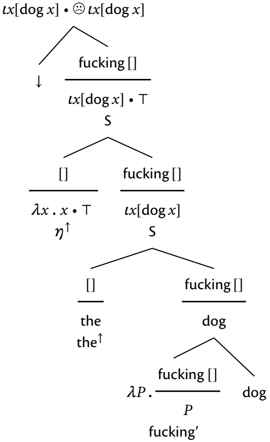
(40) $\text{fucking}'\ m \coloneqq \dfrac{\text{fucking}\ []}{m} \qquad \text{a} \rightarrow \dfrac{\text{W e}}{\text{a}}$

fucking' contributes an identity function locally, and waits for a fancy individual in order to evaluate its scope.

fucking' generalizes our original treatment of *fucking* – the scopal side-effect of *fucking* can be immediately evaluated just in case it composes with a fancy individual.

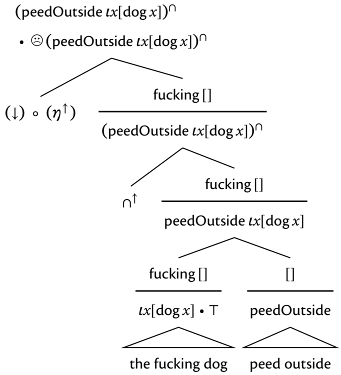


(41) The fucking dog.



Clausal readings

(42) The fucking dog peed outside.



One way of accounting for clausal readings without positing a polysemous *fucking* is to invoke a proposition-to-individual shift.

Hypothesis: the expressive contribution of adjectives such as *fucking* takes scope; so-called "non-local" readings of expressive adjectives are a scopal phenomenon.

Prediction: Non-local readings of expressive adjectives should be sensitive to scope islands.

Scope islands and expressives

Gutzmann (2019) provides extensive argumentation that non-local readings of EAs are subject to syntactic restrictions – they are sensitive to syntactic islands such as relative clauses, but crucially also cannot extend out of finite clauses, just like other scope-takers.

(43) Peter said [that the dog ate the damn cake].
✗☹ (*Peter said that the dog ate the cake*)
✗☹ (*Peter*)

(44) The dog that ate the damn cake is hungry.
✗☹ (*the dog that ate the cake is hungry*)
✗☹ (*the dog that ate the cake*)

The sensitivity of EAs to scope islands falls out as a *prediction* of the semantics we assigned to them.

Consider the semantics of an unevaluated relative clause with an expressive side-effect:

(45) $$[\![\text{[that ate the damn cake]}]\!] = \frac{\text{fucking } [\,]}{\lambda y\,.\,y \text{ ate theCake}} \qquad \frac{\text{W e}}{\text{e} \rightarrow \text{t}}$$

The scope of the expressive cannot be evaluated due since the bottom of the tower isn't (and can't be shifted to) type W e. The scope of the expressive must therefore be evaluated *inside* of the relative clause.

It's important to note that expressive side-effects *once evaluated* are predicted to (and do) survive through scope islands. Consider the semantics of an *evaluated* relative clause with expressive side effects:

(46) $[\![$[that ate the damn cake]$]\!] = \lambda y \,.\, y$ ate theCake • ☹ theCake

The expressive attitude assigned to the cake percolates up via vanilla writer-monadic composition:

(47) $[\![$the dog that ate the damn cake is hungry$]\!]$

$=$ hungry ιx[dog $x \wedge x$ ate theCake] • ☹ theCake

Quantification, binding, and expressives

When uttered by a speaker who likes cats, (48) can express a negative attitude towards whichever cat that happens to be being affectionate – the resolution of the expressive attitudes is therefore *indeterminate*.

(48) A fucking cat is being affectionate for once. ✗ $\exists x[☹ x]$

This would fail to guarantee that the target of the expressive attitude is the same as the cat being affectionate. Rather, it seems like we want the existential quantifier to take scope over *both* the descriptive content and the expressive content.

Something like: $\exists x[(\text{cat } x \wedge \text{affectionate } x) \bullet ☹ x]$

How do we accomplish this compositionally?

By way of contrast:

(49) Every fucking cat is being affectionate for once. ✓$\forall x[☹\, x]$

Charlow (2014) makes use of the **State.Set** monad in order to provide a monadic grammar which captures indefinites' ability to trigger indeterminacy and to introduce discourse referents.

(50) $\mathsf{S\,a} \coloneqq \mathsf{s} \rightarrow \{\langle \mathsf{a}, \mathsf{s}\rangle\}$

(51) $x^{\eta} \coloneqq \lambda i\, \{\langle x, i\rangle\}$

(52) $m \gg= k \coloneqq \lambda i\,.\, \bigcup_{\langle x,j\rangle \in m\, i} k\, x\, j$

(53) aDog $= \lambda i \,.\, \{\, \langle x, s + x \rangle \mid \mathrm{dog}\, x \,\}$

$$\lambda i \,.\, \{\, \langle \mathrm{isOutside}\; x, i + x \rangle \mid \mathrm{dog}\, x \,\}$$

$$\circledast$$

$$\lambda i \,.\, \{\, \langle x, s + x \rangle \mid \mathrm{dog}\, x \,\} \qquad \lambda i \,.\, \{\, \langle \mathrm{isOutside}, i \rangle \,\}$$

a dog — is outside$^{\eta}$

Indefinite phrases trigger state-sensitive, branching computations.

(54) $\mathsf{a} \coloneqq \lambda c \,.\, \lambda s \,.\, \{\langle x, s' \rangle \mid \langle \mathsf{True}, s' \rangle \in c\, x\, s\}$ $(\mathsf{e} \to \mathsf{S}\,\mathsf{t}) \to \mathsf{S}\,\mathsf{e}$

Tripartite type:

$$\frac{\mathsf{S}\,\mathsf{e} \,\big|\, \mathsf{S}\,\mathsf{t}}{\mathsf{e}}$$

Tower meaning:

$$\frac{\mathsf{a}\,(\lambda x \,.\, [])}{x}$$

Composing restrictors with determiners

(55) $\dfrac{\text{a}\,(\lambda x\,.\,[\,])}{x}$ $\qquad$ $\dfrac{\text{S e} \,\big|\, \text{S t}}{\text{e}}$

(56) $\dfrac{[\,]}{\text{dog}}$ $\qquad$ $\dfrac{\text{S t}}{\text{e} \to \text{t}}$

(57) $\dfrac{\text{a}\,(\lambda x\,.\,[\,])}{\text{dog}\ x}$ $\qquad$ $\dfrac{\text{S e} \,\big|\, \text{S t}}{\text{t}}$

(58) $\text{a}(\lambda x\,.\,(\text{dog}\ x)^{\eta})$ $\qquad$ S e

(59) $= \lambda s\,.\,\{\langle x, s\rangle \mid \text{dog}\ x\}$

It's easy to define a new monad, F, which folds the expressive dimension via **Writer** into Charlow's **State.Set** – we can call it **State.Set.Writer**.

(60) $\mathsf{F\,a} \coloneqq \mathsf{s} \to \{\, \langle \mathsf{a} \bullet \mathsf{t}, \mathsf{s} \rangle \,\}$

(61) $x^{\eta} \coloneqq \lambda i \,.\!, \{\, \langle x \bullet \top, i \rangle \,\}$

(62) $$m \ggg\!= k = \lambda i \,.\, \left\{ \langle v \bullet p \wedge q, h \rangle \;\middle|\; \begin{array}{l} \langle x \bullet p, j \rangle \in m\, i \\ \wedge \langle v \bullet q, h \rangle \in k\, x\, j \end{array} \right\}$$

Our first step is to redefine *fucking* as a function from F–fancy individuals to F–fancy individuals.

(63) fucking m $:= \lambda i \,.\, \{ \langle x \bullet p \wedge ☹ x, j \rangle \mid \langle x \bullet p, j \rangle \in m\ i \}$ $\quad$ F e → F e

We then lift the resulting meaning into a scope taker, just as before.

(64) fucking$'$ $m := \lambda k$. fucking $k\ m$ $\quad$ $\mathrm{a} \to \dfrac{\mathrm{F\,e}}{\mathrm{a}}$

Composing this with a predicate such as dog, gives us back a predicate with an expressive side effect that takes scope.

(65) $\dfrac{\text{fucking } [\,]}{\text{dog}}$ $\quad$ $\dfrac{\mathrm{F\,e}}{\mathrm{e} \to \mathrm{t}}$

Q: how do we compose expressions of type $\dfrac{\mathsf{S\,e} \mid \mathsf{S\,t}}{\mathsf{e}}$ and $\dfrac{\mathsf{F\,e}}{\mathsf{e} \to \mathsf{t}}$?

First, we *externally* lift **a**, adding a multi-dimensional tier.

(66) $\mathrm{a}^{\Uparrow} = \dfrac{[]}{\dfrac{\mathrm{a}\,(\lambda x\,.\,[])}{x}}$ $\qquad \dfrac{\mathsf{F\,e}}{\dfrac{\mathsf{S\,e} \mid \mathsf{S\,t}}{\mathsf{e}}}$

Now, we *internally* lift *fucking dog*, adding a **State.Set** tier.

$$\text{(67)}\quad \text{fucking dog}^{\uparrow\uparrow} = \cfrac{\text{fucking []}}{\cfrac{[]}{\text{dog}}} \qquad\qquad \cfrac{\mathsf{F\,e}}{\cfrac{\mathsf{S\,t}}{e \to t}}$$

Now we compose the two via tripartite scopal application:

$$(68)\quad \dfrac{\text{fucking}\,[\,]}{\dfrac{\text{a}\,(\lambda x\,.\,[\,])}{\text{dog}\,x}} \qquad \dfrac{\text{F e}}{\dfrac{\text{S e} \,|\, \text{S t}}{\text{t}}}$$

We can lower the bottom tiers of the tower into a fancy individual in the S dimension.

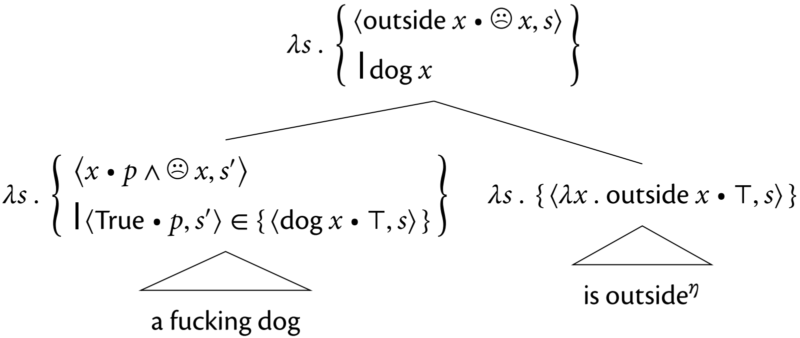
$$(69)\quad \dfrac{\text{fucking}\,[\,]}{\lambda s\,.\,\{\langle x, s\rangle \mid \text{dog}\,x\}} \qquad \dfrac{\text{F e}}{\text{S e}}$$

Finally, we lift the bottom tier of the tower from the **State.Set** to the **State.Set.Writer** dimension.

(70) $$\frac{[]}{\eta^{\eta}}\,\mathsf{S}\,\frac{\text{fucking}\,[]}{\lambda s\,.\,\{\langle x, s\rangle \mid \text{dog}\,x\}} = \frac{\text{fucking}\,[]}{\lambda s\,.\,\{\langle x \bullet \top, s\rangle \mid \text{dog}\,x\}}$$

Finally, we collapse the whole thing via lower.

(71) a fucking dog $= \lambda s\,.\,\{\langle x \bullet ☹\, x, s' + x\rangle \mid \text{dog}\,x\}$


$\lambda s \,.\, \left\{ \begin{array}{l} \langle \text{outside } x \bullet ☹ x, s \rangle \\ \mid \text{dog } x \end{array} \right\}$
$\lambda s \,.\, \left\{ \begin{array}{l} \langle x \bullet p \wedge ☹ x, s' \rangle \\ \mid \langle \text{True} \bullet p, s' \rangle \in \{ \langle \text{dog } x \bullet \top, s \rangle \} \end{array} \right\}$
$\lambda s \,.\, \{ \langle \lambda x \,.\, \text{outside } x \bullet \top, s \rangle \}$
a fucking dog
is outside$^{\eta}$

Expressives and binding

(72) A fucking cat is being affectionate for once.
Its fucking friend is (being affectionate for once) too.

(72) has a reading according to which the speaker has a negative attitude towards an indeterminate cat *x*, and a negative attitude towards *x*'s friend (whoever that turns out to be).

(73)
$$\left\{ \begin{array}{l} \text{affectionate}\ x \\ \wedge\ \text{affectionate}\ y \end{array} \bullet \begin{array}{l} \text{☹}\, x \\ \wedge\ \text{☹}\, y \end{array} \middle| \begin{array}{l} \text{cat}\ x \\ \wedge\ y\ \text{friend}\ x \end{array} \right\}$$

This shows that expressive content must be *state sensitive* in exactly the way that our compositional regime predicts, since the expressive content associated with the second sentence is sensitive to the discourse referent introduced in the first.

I won't show the details here, but we predict that any dynamically closed operator that block the DP-level reading of an expressive with an indefinite – just in case the scope of the indefinite is roofed by negation.

(74) I didn't see a fucking dog.

If the indefinite is interpreted within the scope of negation, this *can't* be interpreted as an expressive attitude targetting any particular dog.

To the extent that expressives allow DP-level readings with other quantifiers, we predict that the expressive attitude should always target a maximal dref. This seems correct.

(75) Hans gave a good grade to exactly three fucking students.

Crucially, this doesn't convey that the speaker has a negative attitude towards exactly three students – this is way too weak, rather it conveys, for whichever students X Hans gave a good grade to, the speaker has a negative attitude towards X.

Conclusion

Conclusion

- Following work by Giorgolo & Asudeh (2012) and Charlow (2015), we've shown that *expressive content* can be integrated into an existing compositional regime, in a fully compositional way, via **Writer**.
- Marshalling independently motivated mechanisms for scope-taking (i.e. the **Cont** monad), opens the doors to a compositional account of *non-local readings* of EAs.
- Once we integrate **Writer** into a monadic grammar with the resources to model *state-sensitivity* and *indeterminacy*, we end up with a system which accounts – without much else needing to be said – for the distinctive way in which *expressive content* interacts with logical operators such as quantificational determiners.

Thank you

Acknowledgements

Thanks especially to Elin McCready and Ryan Walter Smith for much useful discussion.

References

Barker, Chris & Chung-chieh Shan. 2014. *Continuations and natural language.* (Oxford studies in theoretical linguistics 53). Oxford University Press. 228 pp.

Charlow, Simon. 2014. *On the semantics of exceptional scope.*

Charlow, Simon. 2015. Conventional implicature as a scope phenomenon. Slides from an invited talk at the workshop on continuations and scope. NYU.

Elliott, Patrick D. 2019. Applicatives for anaphora and presupposition. To appear in the Proceedings of LENLS15.

Giorgolo, Gianluca & Ash Asudeh. 2012. $\langle M, \eta, * \rangle$: monads for conventional implicatures. In *Proceedings of sinn und bedeutung 16* (MIT Working papers in Linguistics), 265–278.

Gutzmann, Daniel. 2013. Expressives and beyond: an introduction to varieties of use-conditional meaning. In Daniel Gutzmann & Hans-Martin Gärtner (eds.), *Beyond expressives: explorations in use-conditional meaning*, 1–58. Brill.

Gutzmann, Daniel. 2019. *The grammar of expressivity*. New York, NY: Oxford University Press.

May, Robert. 1977. *The grammar of quantification*. Massachussetts Institute of Technology dissertation.

McCready, E. 2010. Varieties of conventional implicature. *Semantics and Pragmatics* 3.

McCready, E. 2012. Emotive equilibria. *Linguistics and Philosophy* 35(3). 243–283.

Shan, Chung-chieh. 2002a. A continuation semantics for interrogatives that accounts for Baker's ambiguity. In Brendan Jackson (ed.), *Salt xii*, 246–265. Massachussetts Institute of Technology: Linguistic Society of America.

Shan, Chung-chieh. 2002b. Monads for natural language semantics. *arXiv:cs/0205026*.