

Categories of Semantic Concepts

James Hefford¹, Vincent Wang¹, and Matthew Wilson^{1,2}

¹University of Oxford, Department of Computer Science, Quantum Group

²HKU-Oxford Joint Laboratory for Quantum Information and Computation

April 23, 2020

Abstract

Modelling concept representation is a foundational problem in the study of cognition and linguistics. This work builds on the confluence of conceptual tools from Gärdenfors semantic spaces, categorical compositional linguistics, and applied category theory to present a domain-independent and categorical formalism of ‘concept’.

1 Introduction

Gärdenfors proposes geometry as a mediator between raw empirical data and symbolic cognitive representations of concepts [11]. Conceptual spaces conceived in this way were instrumental in developing state-of-the-art models for semantic inference from linguistic data [4], in conjunction with tools from Firth’s distributional models of meaning, and the DisCoCat programme of categorial grammar and semantics [8].

Gärdenfors’ model treats concepts as regions of abstract geometric domains that encode properties; for instance, where the domains are colour and taste, the concept ‘Banana’ might be expressed as a well-behaved restricted region that captures the expected inferences that no bananas are red and sour, while yellow bananas tend to be sweet, and green ones bitter.

We develop towards a categorical formalism that builds categories of semantic concepts – such as ‘Banana’ – on top of arbitrary categories whose objects are the modeller’s choice of conceptual spaces that constitute what Gärdenfors calls *integral domains*. We briefly recount Gärdenfors’ ontology. An integral domain is a semantic space where one cannot have incomplete information, just as one cannot hear the pitch of a sound without also perceiving the volume. Colourspace (more accurately, Hue) is such an integral domain of three quality dimensions, redness, greenness, and blueness. Similarly, tastespace is an integral domain, along, say, the dimensions of sweetness, savouriness, bitterness, sourness, and umami. However, when considering fruitsspace, modelled as the (tensored) conjunction of colour and taste, it is possible to specify colour without taste (by just looking), and taste without colour (by eating blindfolded.) In this case we refer to the domains of colour and taste as independent domains¹. We abstractly deem a concept to be a subregion of possibly

¹‘integral’ and ‘separable’ are also referred to as ‘interacting’ and ‘separable’ in the literature

multiple tensorially conjoined independent domains. For instance, where fruitspace is the tensor of colourspace and tastespace, the concept ‘banana’ falls within an acceptable range of yellowness and greenness, sweetness and bitterness, in a potentially non-trivial way: we may tolerate different taste-ranges in bananas depending on their colour. Percepts that stray too far outside these constraints may not be intuitively classified as bananas *per se*, as sour purple bananas are like black swans: arguably not bananas at all.

In the remainder of this paper, we first briefly introduce Symmetric Monoidal Categories and their attendant graphical calculus, the tool of choice for the DisCoCat school of modelling categorical semantics. We then develop a cognitively-driven axiomatic account of updating static properties of systems. We then introduce the notion of correlators on update structures, and by demonstration, we exhibit how concepts may be constructed from domains by their restriction and correlating actions. We then recover from these tools the first categorical formulation of Gärdenfors’ semantic concepts, and show how our construction reconciles with our account of updating.

2 Mathematical Background

For a thorough account of basic category theory, we refer the reader to [1]. For an account of the available graphical calculi for various monoidal categories, we refer to [14].

Definition 1 (Category). A category \mathbf{C} consists of the following data:

- A collection $\text{Ob}(\mathbf{C})$, the elements of which we call **objects** of \mathbf{C} .
- For every pair of objects c, d , a set $\mathbf{C}(c, d)$, elements of which are called **morphisms**, or **arrows**, from c to d . We might denote such a morphism $f : c \rightarrow d$, or $c \xrightarrow{f} d$.
- For every object c , a specification of a morphism $\text{id}_c \in \mathbf{C}(c, c)$, called the **identity morphism** on c .
- For every triple of objects c, d, e , and every pair of morphisms $f \in \mathbf{C}(c, d)$ and $g \in \mathbf{C}(d, e)$, a specification of a morphism in $\mathbf{C}(c, e)$ called the **composite** of f and g , denoted either $g \circ f$, or $f; g$. The former is read “ g after f ”, and the latter is read “ f , then g ”.

Subject to the following conditions:

- (*Unitality*): For any $c \xrightarrow{f} d$, we have $\text{id}_c; f = f = f; \text{id}_d$; that is, composing with identities does nothing.
- (*Associativity*): For any three morphisms $a \xrightarrow{f} b$, $b \xrightarrow{g} c$, $c \xrightarrow{h} d$, we have $(f; g); h = f; (g; h)$.

Definition 2 (Monoidal Category). A monoidal category is a tuple $(\mathbf{C}, \otimes, I, \alpha, \lambda, \rho)$ where:

- \mathbf{C} is a category
- \otimes , the **tensor**, is a functor $\mathbf{C} \times \mathbf{C} \rightarrow \mathbf{C}$, and we use the infix notation $A \otimes B$ for $\otimes(A, B)$

- I , the **unit**, is an object of \mathbf{C}
- α, λ, ρ are families of natural isomorphisms, with types (for all X, Y, Z in \mathbf{C}):
 - $\alpha_{X,Y,Z} : ((X \otimes Y) \otimes Z) \rightarrow (X \otimes (Y \otimes Z))$ – the **associator**
 - $\rho_X : X \otimes I \rightarrow X$ – the **right unitor**
 - $\lambda_X : I \otimes X \rightarrow X$ – the **left unitor**

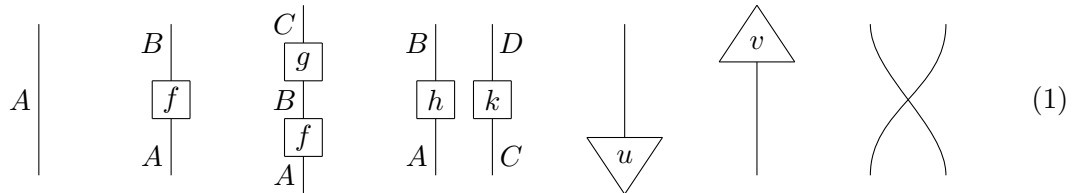
Further, these natural isomorphisms must be such that any well-typed diagram consisting of $\otimes, \alpha, \lambda, \rho, \alpha^{-1}, \rho^{-1}, \lambda^{-1}$ commutes.

A monoidal category is further **symmetric** if there is a **braiding** natural transformation $\theta_{X \otimes Y} : X \otimes Y \rightarrow Y \otimes X$ such that

$$\theta_{Y \otimes X} \circ \theta_{X \otimes Y} = \text{id}_{X \otimes Y}$$

Monoidal Categories are models of process theories [6], and admit a sound and complete graphical calculus. The elements of the graphical calculus are as follows, from left to right of (1):

- A wire of type A (the identity process on the object A), which ferries stuff of that type.
- A process named f , drawn as an axis-aligned box, which takes stuff of type A as input, and outputs stuff of type B .
- A vertical composite process of named processes f and g . First f ‘transforms’ A -stuff to B -stuff, and then g transforms that B -stuff into C -stuff.
- A horizontal composite process of named processes h and k . Since we can name h and k separately, why not invoke those processes concurrently? This composite takes A -stuff and C -stuff, and returns B -stuff and D -stuff.
- A process u that gives some output, but takes no input. We call these ‘states’.
- A process v that accepts some input, but gives no output. We call these ‘effects’.
- The braid natural isomorphism is depicted as a pair of crossing wires.



The identity on the monoidal unit I is represented as a blank diagram. In this work, we read diagrams bottom to top. In symmetric monoidal categories (SMCs), two diagrams are equal if they can be deformed into one another while keeping all boxes and lines on the page, that is to say they are equivalent up to planar isotopy. For example:

(2)

Since we take our braiding to be symmetric, one need not worry about the order in which wires cross each other. Nevertheless, most of this work can be easily extended to the case of non-symmetric braided monoidal categories.

The commonality of extant approaches to modelling semantics in categorial grammar is that wires are interpreted as ‘meaning spaces’, and states on such wires correspond to specific objects. For instance, in the category **Set** of sets and functions, viewed as a SMC with cartesian product as monoidal product and singletons as monoidal unit is a monoidal category, the wires are sets, and states on wires are specific elements of those sets. In the SMC **Vect** of vector spaces and linear maps with \otimes as tensor product, states on wires are vectors, which are used to encode semantics in distributional models of meaning.

3 Axiomatic Updating

In the following, we borrow terminology from the language of lenses, as studied in applied category theory. Lenses were developed as formal constructs [10] to study synchronised update systems [3], and our aims here are broadly aligned. Our basic object of study is what we call an ‘update structure’, which models updating an abstract system with an abstract property. We define what qualifies as an update structure in an SMC axiomatically, reflecting three intuitive cognitive criteria for updating static properties of systems.

1. Examining a property and then immediately overwriting with the same property is the trivial operation.
2. Updating an object with properties in succession is procedurally equivalent to manipulating the collection of properties in some way beforehand to obtain a single property, and then overwriting once.
3. Repeatedly examining a property produces the same outcomes.

Definition 3 (Update Structure). An update structure $(\uparrow, \odot, \sim, \downarrow)$ in a SMC consists of:

- An object S , which we refer to as a **system**
- An object p , which we refer to as a **property**, which has:

- A magma structure $\sim: p \otimes p \rightarrow p$

- A cocommutative, coassociative comagma structure $\curlyvee : p \rightarrow p \otimes p$
- A **Put** operation $\uparrow : S \otimes p \rightarrow S$
- A **Get** operation $\blacktriangleright : S \rightarrow S \otimes p$

Which satisfy the following equations:

- (PutGet) To capture the intention of overwriting:

$$(3)$$

- (GetPut) To capture our first criterion, treating the identity process as trivial:

$$(4)$$

- (PutPut) To capture our second criterion, using the \sim magma to encode property pre-processing:

$$(5)$$

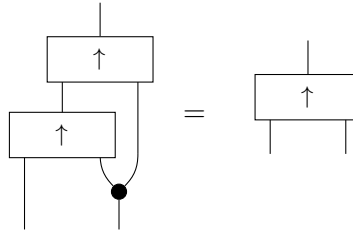
- (GetGet) To capture our third criterion, following from the cocommutativity of \curlyvee :

$$(6)$$

It should be noted that axiom (5) says that \uparrow is a “weak” module over the magma \sim while (6) says that \odot is a “weak” comodule over the comagma \smile . By weak we mean that they have no (co)units, nor are they actions over a (co)monoid as would be more usual. Modules and comodules in their usual setting have seen application in the categorical quantum mechanics line of research where they describe observables [7].

We also define a slight weakening of an update structure, one where the GetPut axiom does not hold, but where it is replaced with the *repeat-update* axiom.

Definition 4 (Weak Update Structure). A weak update structure $(\uparrow, \odot, \sim, \smile)$ satisfies all the axioms of an update structure apart from GetPut (4) which we replace with the strictly weaker *repeat-update* axiom:


(7)

A weak update is one where the Put is static, in the sense that updating twice with copies of your input is the same as updating once.

Proposition 1. *Any update structure is also a weak update structure.*

Proof. Follows by application of axiom 3 followed by axiom 4 to the right hand side of 7, given explicitly in the appendix. \square

In section 6 we will show how to “upgrade” any weak update structure to an update structure.

Given the semantic modeller’s choice of SMC, we take the objects of that SMC to be conceptual spaces. For instance, in the SMC of convex spaces and convexity preserving maps between them, there is an object corresponding to RGB colour-space, the states of which are particular colours. Conceptual spaces admit concepts as subregions, as ‘redness’ would be defined as a subregion of RGB colour space. Generally, such concepts may be complex subregions of conceptual spaces which do not have an analogue object in the original SMC. Moreover, concepts like ‘monkeyness’ might be modelled in terms of other, hierarchical concepts, such as ‘emotion’ and ‘banananness’, where the latter is modelled in terms of primitive conceptual spaces such as ‘colour’ and ‘taste’.

We may define concepts from conceptual spaces through update structures, by way of what we call ‘semantic correlators’. Considering concepts as semantic correlators along with update structures provides the necessary conceptual tools to construct complex hierarchical concepts such as ‘monkeyness’ from other concepts.

4 Semantic Correlators

Given update structures for multiple different properties, an update structure for colour and an update structure for taste say, it is natural to ask how one can combine these together to create new update structures. In particular, one might wish to build a new update structure in the presence of additional information about how two properties combine together. For instance, if one were considering bananas then it is typical that yellow bananas are sweet and green bananas are bitter and the properties of colour and taste are no longer fully independent. To capture such instances we introduce a new map we call the correlator.

Definition 5 (Correlator). Given a family of properties $A_1, \dots, A_n, n \in \mathbb{N}$, a correlator is an idempotent map $\mathfrak{X} : A_1 \otimes \dots \otimes A_n \rightarrow A_1 \otimes \dots \otimes A_n$, which is a comagma homomorphism for the comagma induced by those on each A_i ,

(8)

and a magma homomorphism for the magma induced by those on each A_i

(9)

One should think of the correlator as containing additional information about how certain properties influence each other. As such, correlators are very much dependent on the new concept one wishes to create from the underlying ones; the correlator for a banana is clearly very different from that of a fire engine, for instance.

The idempotence of a correlator captures the notion that properties correlated in a particular fashion are invariant under correlating them the same way again. Demanding that a correlator be a comagma and magma homomorphism captures the notion that copying (mixing) then correlating ought to be the same as correlating then copying (mixing).

For simplicity we will now work with correlators on only two properties, but the following techniques can easily be extended to more complicated cases. Suppose we have two properties, c and t with corresponding update structures $(\uparrow_c, \odot_c, \sim_c, \downarrow_c)$ and $(\uparrow_t, \odot_t, \sim_t, \downarrow_t)$ on a system S and suppose that the structures commute, that is $\uparrow_t \circ \uparrow_c = \uparrow_c \circ \uparrow_t$ and $\odot_t \circ \odot_c = \odot_c \circ \odot_t$ where the composition is on the conceptual space S and equality holds up to a braid on the properties². Then one can construct a new correlated Put \uparrow_{ct} and correlated Get \odot_{ct} by

²This notion of the *independence* of the properties c and t reflects Gärdenfors' notion of independent domains, and is not unreasonable to require from a modelling perspective.

taking

$$\begin{array}{ccc}
 \begin{array}{c} S \\ \downarrow \\ \boxed{\uparrow_{ct}} \\ \downarrow \\ S \quad c \quad t \end{array} & = & \begin{array}{c} S \\ \downarrow \\ \boxed{\uparrow_t} \\ \downarrow \\ \boxed{\uparrow_c} \\ \downarrow \\ S \quad c \quad t \end{array} \quad , \quad \begin{array}{c} S \quad c \quad t \\ \downarrow \\ \boxed{\text{eye}_{ct}} \\ \downarrow \\ S \end{array} = \begin{array}{c} S \quad c \quad t \\ \downarrow \\ \boxed{\text{eye}_c} \\ \downarrow \\ \boxed{\text{eye}_t} \\ \downarrow \\ S \end{array}
 \end{array} \quad (10)$$

Such a choice nearly forms a weak update structure but sadly PutGet only holds up to the correlator on input properties:

$$\begin{array}{c} \text{eye}_{ct} \\ \downarrow \\ \boxed{\uparrow_{ct}} \end{array} = \begin{array}{c} \boxed{\uparrow_{ct}} \\ \downarrow \\ \bullet \quad \bullet \\ \downarrow \\ \boxed{\text{X}} \end{array}$$

We will be able to resolve this issue by moving the the category of semantic concepts in section 5.

For clarification of how the process of constructing correlators and correlated updates works consider the aforementioned example regarding the colour and taste of bananas. Suppose we identify the possible colours with the set $\llbracket \text{colour} \rrbracket := \{\text{yellow}, \text{green}\}$ and possible tastes with the set $\llbracket \text{taste} \rrbracket := \{\text{bitter}, \text{sweet}\}$ and assume for simplicity that yellow bananas are always sweet and green bananas always bitter. Thus we have a map $P : \llbracket \text{colour} \rrbracket \rightarrow \llbracket \text{taste} \rrbracket :: \text{yellow} \mapsto \text{sweet}, \text{green} \mapsto \text{bitter}$. Take the comultiplication on colour to be the copying map of the elements of the set of colours, $\curlyvee : i \mapsto (i, i)$ for $i \in \llbracket \text{colour} \rrbracket$, and similarly for taste. One can then define the following correlator

$$\boxed{\text{X}} = \begin{array}{c} \begin{array}{c} \downarrow \\ \text{P} \\ \downarrow \\ \circ \\ \downarrow \\ \text{P}^{-1} \\ \downarrow \end{array} \end{array} \quad (11)$$

which is clearly idempotent and it is straightforward to check that it is a comonoid homomorphism for the comonoid induced by copying colour and taste.

Now take the following Gets, Puts and Mixes:

$$\begin{array}{ll}
 \uparrow_c :: \langle c, t \rangle \times c' \mapsto \langle c', t \rangle & \uparrow_t :: \langle c, t \rangle \times t' \mapsto \langle c, t' \rangle \\
 \text{eye}_c :: \langle c, t \rangle \mapsto \langle c, t \rangle \times c & \text{eye}_t :: \langle c, t \rangle \mapsto \langle c, t \rangle \times t \\
 \sim_c :: c \times c' \mapsto c' & \sim_t :: t \times t' \mapsto t'
 \end{array}$$

for $c, c' \in \llbracket \text{colour} \rrbracket$ and $t, t' \in \llbracket \text{taste} \rrbracket$.

Then one can confirm that X (11) is a monoid homomorphism for the monoid induced by \sim_c and \sim_t . Furthermore, it is possible to construct the correlated weak update where green bananas are bitter and yellow bananas sweet from the above data.

5 Fruit Wires: The Category of Semantic Concepts

Taking the view that concepts can be defined by the correlations they exhibit between properties in independent domains, we aim to construct the concept *banana* (🍌) from the domains *colour* (c) and *taste* (t) along with the correlator

$$\mathfrak{K}_{\text{🍌}} : c \otimes t \rightarrow c \otimes t$$

Informally, any state ψ on the property pair $c \otimes t$ which is consistent with the correlator should represent an instance of a banana:

$$\begin{array}{c} \text{---} \\ | \\ \boxed{\mathfrak{K}_{\text{🍌}}} \\ | \\ \triangle \psi \end{array} = \begin{array}{c} \text{---} \\ | \\ \triangle \psi \end{array} \quad (12)$$

Additionally the processes that can be performed on bananas should be all of those which are consistent with the correlations and applying the correlator to the inputs and outputs of f should have no effect:

$$\begin{array}{c} \text{---} \\ | \\ \boxed{f} \\ | \\ \boxed{\mathfrak{K}_{\text{🍌}}} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \boxed{f} \\ | \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ | \\ \boxed{\mathfrak{K}_{\text{🍌}}} \\ | \\ \boxed{f} \\ | \\ \text{---} \end{array} \quad (13)$$

One can capture such scenarios by transitioning from working in the category \mathbf{C} to the Karoubi envelope $\bar{\mathbf{C}}$.

Definition 6 (Karoubi Envelope). The Karoubi envelope $\bar{\mathbf{C}}$ of a category \mathbf{C} has as objects the pairs (A, π) where $A \in \text{Ob}(\mathbf{C})$ and $\pi : A \rightarrow A$ is an idempotent. The morphisms $f : (A, \pi) \rightarrow (B, \sigma)$ are the morphisms $f : A \rightarrow B$ such that $\sigma \circ f = f = f \circ \pi$.

Objects of the form $(c \otimes t, \mathfrak{K}_{\text{🍌}})$ in the Karoubi envelope $\bar{\mathbf{C}}$ are precisely those for which the only allowed processes are those consistent with $\mathfrak{K}_{\text{🍌}}$ and thus it makes sense to define new concepts as objects of this form for a given correlator.

Definition 7 (Concept). The concept corresponding to the correlator $\mathfrak{K} : (A, \pi_A) \otimes (B, \pi_B) \rightarrow (A, \pi_A) \otimes (B, \pi_B)$ is the object $(A \otimes B, \mathfrak{K})$ in $\bar{\mathbf{C}}$.

We interpret the Karoubi envelope $\bar{\mathbf{C}}$ to be the category of semantic concepts, built from a modelling category of basic concepts \mathbf{C} . It is straightforward to show that there is an equivalence of categories between the full subcategory of $\bar{\mathbf{C}}$ spanned by objects of the form (A, id_A) for all $A \in \text{Ob}(\mathbf{C})$ and the original category \mathbf{C} . Thus all of our initial conceptual spaces appear in $\bar{\mathbf{C}}$ alongside many new derived concepts given by the action of correlators. This allows us treat new concepts as formally different types within our category. Such

notions have also been explored in the categorical quantum mechanics line of research where idempotent decoherence maps can be used to give quantum and classical data different types [9, 13, 12].

An example of a non-basic concept is the *banana* object $\text{banana} = (c \otimes t, \mathfrak{X}_{\text{banana}})$, built from a correlator

$$\mathfrak{X}_{\text{banana}} : (c, \text{id}_c) \otimes (t, \text{id}_t) \rightarrow (c, \text{id}_c) \otimes (t, \text{id}_t)$$

upon the (basic) conceptual spaces *colour* (c, id_c) and *taste* (t, id_t) . Intuitively *banana* wires are *colour* and *taste* wires which have been correlated so as to be consistent with the concept *banana*.

$$\left| \begin{array}{c} \text{banana} \end{array} \right| = \left| \begin{array}{c} (c \otimes t, \mathfrak{X}_{\text{banana}}) \end{array} \right| \approx \begin{array}{c} \begin{array}{|c|c|} \hline c & t \\ \hline \mathfrak{X}_{\text{banana}} \\ \hline c & t \\ \hline \end{array} \end{array} \quad (14)$$

This construction iterates, allowing definition of derived concepts which admit other derived concepts as properties. A further correlation between the flavour (and thus colour) of bananas and the happiness of a consumer induces its own correlator:

$$\mathfrak{X}_{\text{monkey}} : (\text{banana}, \mathfrak{X}_{\text{banana}}) \otimes (\text{happy}, \mathfrak{X}_{\text{happy}}) \rightarrow (\text{banana}, \mathfrak{X}_{\text{banana}}) \otimes (\text{happy}, \mathfrak{X}_{\text{happy}}) \quad (15)$$

which acts on the properties *banana* and *emotion*. Such a correlator could be used to construct the concept *monkey* $\text{monkey} \equiv (\text{banana} \otimes \text{happy}, \mathfrak{X}_{\text{monkey}})$ as the collection of maps which are consistent with the new correlator (15). The consistency of states and processes on monkey with its constituent properties banana and happy is guaranteed since for any $f : \text{monkey} \rightarrow \text{monkey}$,

$$\begin{aligned} f \circ (\mathfrak{X}_{\text{banana}} \otimes \mathfrak{X}_{\text{happy}}) &= f \circ \mathfrak{X}_{\text{monkey}} \circ (\mathfrak{X}_{\text{banana}} \otimes \mathfrak{X}_{\text{happy}}) \\ &= f \circ \mathfrak{X}_{\text{monkey}} = \mathfrak{X}_{\text{monkey}} \circ f = (\mathfrak{X}_{\text{banana}} \otimes \mathfrak{X}_{\text{happy}}) \circ \mathfrak{X}_{\text{monkey}} \circ f = (\mathfrak{X}_{\text{banana}} \otimes \mathfrak{X}_{\text{happy}}) \circ f \end{aligned}$$

We may treat banana as a property of a system (S, id_S) by replacing updates for the basic properties *colour*, and *taste* with an update for the property *banana* $\text{banana} = (c \otimes t, \mathfrak{X}_{\text{banana}})$ on (S, id_S) ; this expresses that (S, id_S) can only have properties c and t consistent as a result of an update of the property banana :

$$\begin{array}{c} S \\ | \\ \boxed{\uparrow_{ct}} \\ | \quad | \quad | \\ S \quad c \quad t \end{array} \cong \begin{array}{c} (S, \text{id}_S) \\ | \\ \boxed{\uparrow_{\text{banana}}} \\ | \quad | \\ (S, \text{id}_S) \quad \text{banana} \end{array} \quad (16)$$

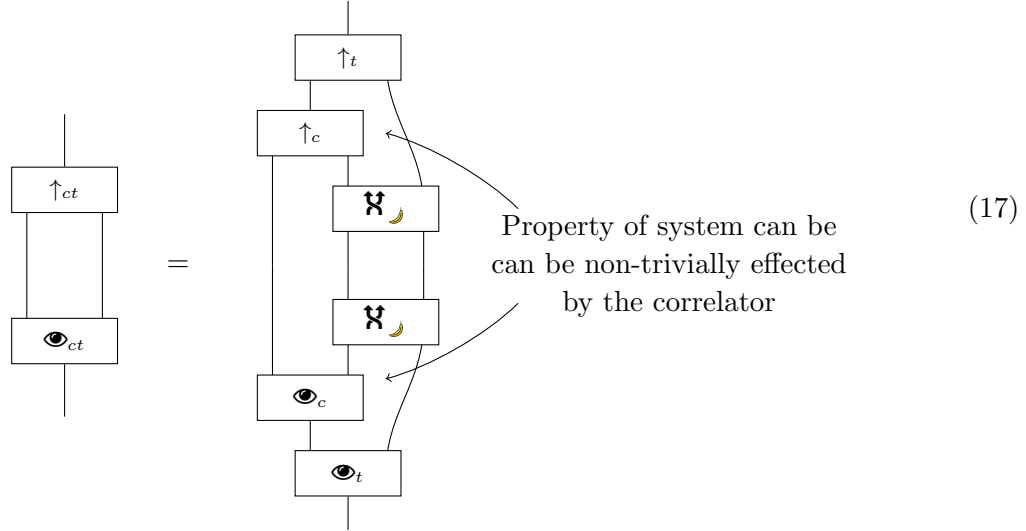
Now that we have developed a way of constructing new concepts from old ones we are able to return to the problem of combining update structures and state the following proposition:

Proposition 2. *Let $(\uparrow_c, \odot_c, \sim_c, \vee_c)$ and $(\uparrow_t, \odot_t, \sim_t, \vee_t)$ be commuting update structures and \mathfrak{X} be a correlator on c and t . Then $(\uparrow_{ct}, \odot_{ct}, \sim_{ct}, \vee_{ct})$ forms a weak update structure on correlated inputs $(c \otimes t, \mathfrak{X})$ where \uparrow_{ct} and \odot_{ct} are defined as in (10) and \sim_{ct} and \vee_{ct} are the magma and comagma induced by those on c and t .*

6 From Weak to Strong Update Structures

We now demonstrate how use update structures to formally enforce that an arbitrary concept, viewed as a system (S, π_S) , has properties c and t consistent with a (derived) property $(b, \pi_b) = (c \otimes t, \mathfrak{X})$, by constructing from a weak update $(\uparrow_b, \bullet_b, \sim_b, \downarrow_b)$ of the property (b, π_b) on the system (S, π_S) a new system (S, π'_S) .

The *repeat-update* axiom, satisfied by the correlated update defined in (10), is a statement of static-ness of the update property. Updating with ‘copies’ of a property b produced by the comagma is the same as updating with that property once. The GetPut axiom $\uparrow_b \circ \bullet_b = \text{id}_{(S, \pi_S)}$ is (in the presence of axioms 3,5,6) strictly stronger than the *repeat-update* axiom; intuitively it is the additional claim that the system (S, π_S) already has its *colour* and *taste* properties consistent with the correlator \mathfrak{X} . The failure of GetPut on a correlated update as defined in (10) comes about because the system wire (S, π_S) is not guaranteed to have *banana-like colour* and *taste* properties.



An informal way to enforce that properties c and t of (S, π_S) are correlated is to apply the GetPut $\uparrow_b \circ \bullet_b$ to anything on (S, π_S) . After retrieval by \bullet_b , the \uparrow_b will apply \mathfrak{X} , thereby correlating the properties c and t . An implication of the *repeat-update* axiom is that $\uparrow_b \circ \bullet_b$ is itself an idempotent,

$$\uparrow_b \circ \bullet_b \circ \uparrow_b \circ \bullet_b = \uparrow_b \circ \bullet_b$$

and thus applying $\uparrow_b \circ \bullet_b$ to a system which is already consistent with the correlations \mathfrak{X} will have no effect. The new object in $\bar{\mathbf{C}}$ which enforces consistency of the system with \mathfrak{X} is then

$$(S, \pi'_S) \equiv (S, \uparrow_b \circ \bullet_b)$$

Any state or process on (S, π'_S) is consistent with the system already having c and t correlated by \mathfrak{X} . What remains is to show that (S, π'_S) itself can actually be practically used: a new update property with system (S, π'_S) and property $(b, \pi_b) = (c \otimes t, \mathfrak{X})$ which satisfies the GetPut axiom $\uparrow_b \circ \bullet_b = \text{id}_{(S, \pi'_S)}$ can be constructed.

Proposition 3 (GetPut Restriction). *For a weak update structure $(\uparrow_b, \textcircled{b}, \sim_b, \downarrow_b)$ on system (S, π_S) and property (b, π_b) satisfying the repeat-update axiom, the GetPut-Restricted updates*

$$(18)$$

along with (\sim_b, \downarrow_b) define an update structure on system $(S, \uparrow_b \circ \textcircled{b})$ and property (b, π_b) .

Proof. By idempotency of $\uparrow_b \circ \textcircled{b}$ it is immediate that \textcircled{b}' and \uparrow'_b appear as morphisms

$$\textcircled{b}' : (S, \uparrow_b \circ \textcircled{b}) \rightarrow (S, \uparrow_b \circ \textcircled{b}) \otimes (b, \pi_b)$$

and

$$\uparrow'_b : (S, \uparrow_b \circ \textcircled{b}) \otimes (b, \pi_b) \rightarrow (S, \uparrow_b \circ \textcircled{b})$$

respectively. The update structure axioms 3,4,5,6 all hold, the proofs of which are given in the appendix. \square

The statement and proof of proposition 3 makes no explicit reference to correlations, and so gives a general way to force consistency of a system with its properties by moving from an update mechanism satisfying the *repeat-update* axiom, to an update mechanism satisfying the GetPut axiom.

7 Conclusion

We have demonstrated a proof-of-concept for the categorical representation of concepts as restricted and correlated regions of Gärdenfors' conceptual spaces, which, to our knowledge, is the first semantic-engineering contribution of this kind.

The core element of our construction is an interacting magma module and comagma comodule, internal to a symmetric monoidal category, reflecting 'update' and 'examine' properties. Though we borrowed terminology from lenses, our approach is distinct on two counts. Firstly, we define our constructs in arbitrary symmetric monoidal categories, whereas lenses typically assume cartesian categories. Secondly, the notion of interacting (co)magma modules is conceptually minimal, there being no analogue in the field of profunctor optics at large to the update structures we have defined, to the best of our knowledge. The benefit of this

conceptual minimalism is that the requirements for update structures to exist are readily met by most approaches to categorical linguistic models. In this work, we have only considered **Set**-based semantic models, but there is no obstacle in principle to application to applying the same approach in true convex geometric domains and vector spaces.

It remains for future work to demonstrate practical applications, say, in terms of inference in vector-based models of linguistic meaning, where we expect our update formalism will permit dynamic epistemic modelling as is encountered in the reading of a long text [5], where the properties of various actors change over time. It is also pressing work to integrate our approach with extant research on density-matrix based models of compositional cognition [2].

A further added benefit of such a conceptually minimal and axiomatic approach is that new domains of semantic models are now amenable for use, for instance, those obtained by RDF triples: no matter the underlying representations, once one has verified that a candidate update structure satisfies the axioms, one can be confident that the structure suitably models updating and viewing properties.

We are surprised by the power and fruitfulness of adopting graphical axiomatics that reflect cognitive modelling intentions, and we believe that further progress can be made in semantic-engineering following this general method. The authors have explored various other axiomatisations of update structures, each capturing different operational intuitions, and yielding qualitatively different systems: it remains for future work to tame the zoo of update structures obtained in this way.

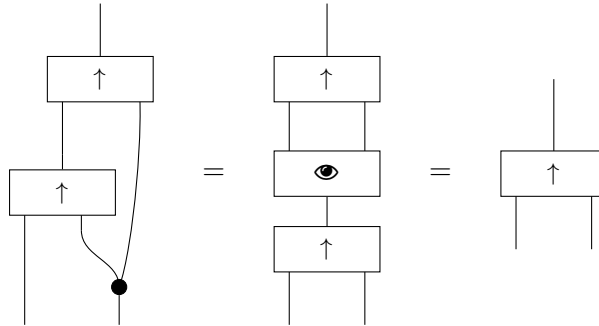
Acknowledgements

With thanks to Sean Tull, Cole Comfort, Irene Rizzo, and Bob Coecke for helpful discussions. This work was supported by University College London and the EPSRC [grant number EP/L015242/1].

A Proofs

A.1 Proof of Proposition 1

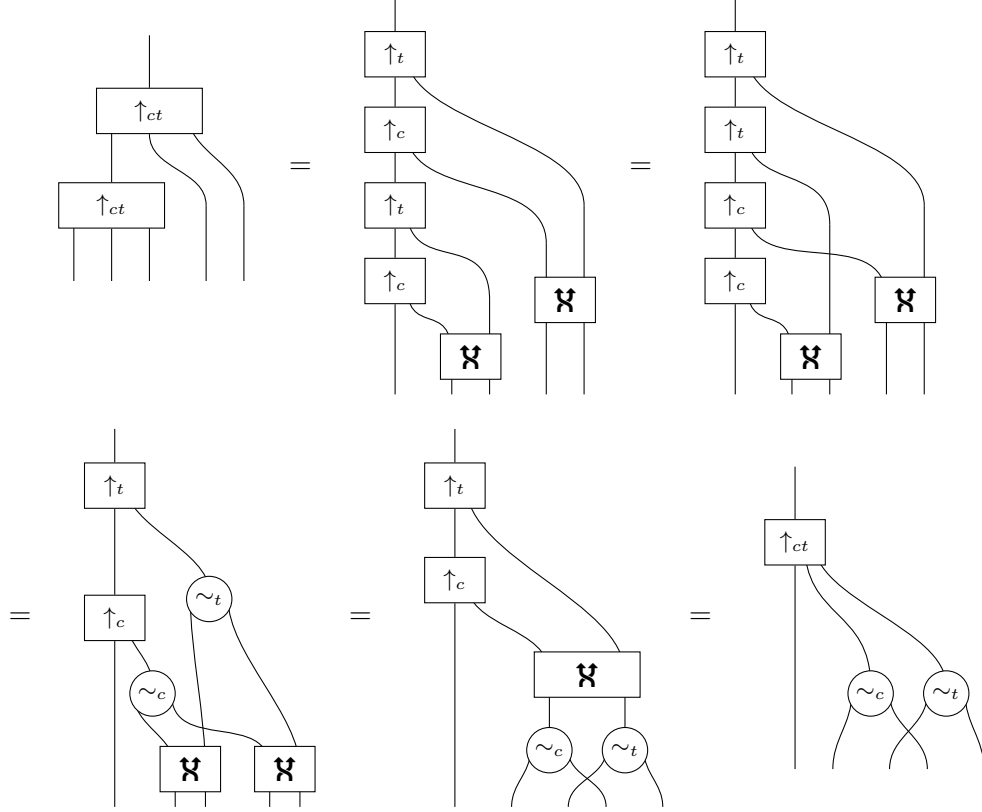
Proof. By application of PutGet followed by GetPut



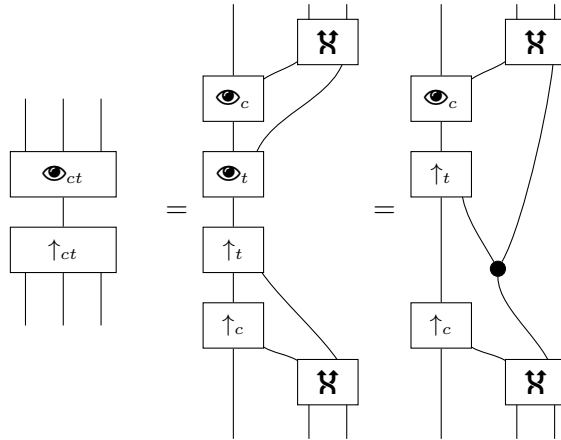
□

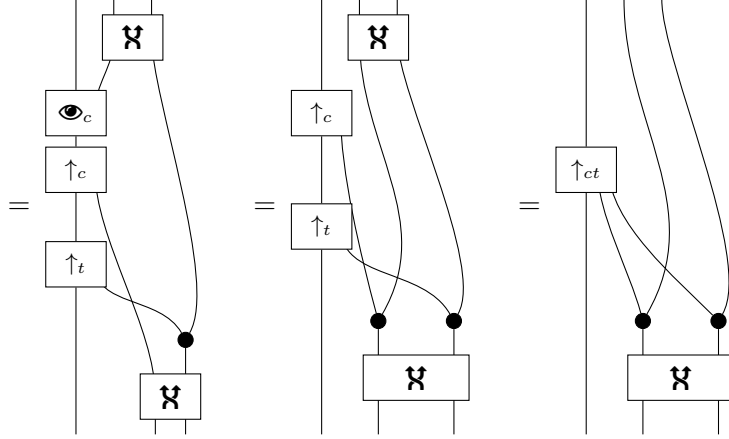
A.2 Proof of Proposition 2

Proof. We check the PutPut, GetGet, PutGet and *repeat-update* axioms, recalling that we assume that the Gets and Puts of the two systems respectively commute. PutPut follows by:

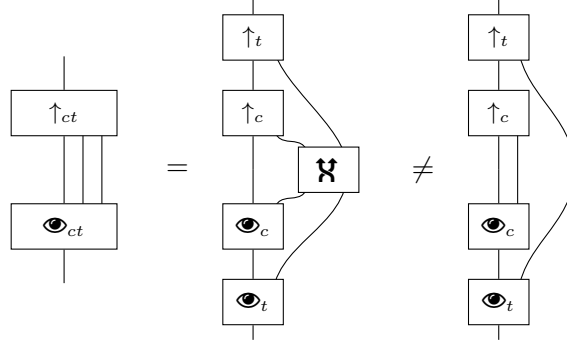


with GetGet very similar. PutGet follows up to correlating the input properties:





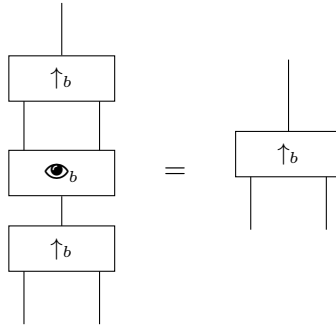
That is to say that we are constructing an update on the new correlated type $(c \otimes t, \mathfrak{X})$ in $\bar{\mathbf{C}}$. Finally, GetPut, does not hold on all input systems:



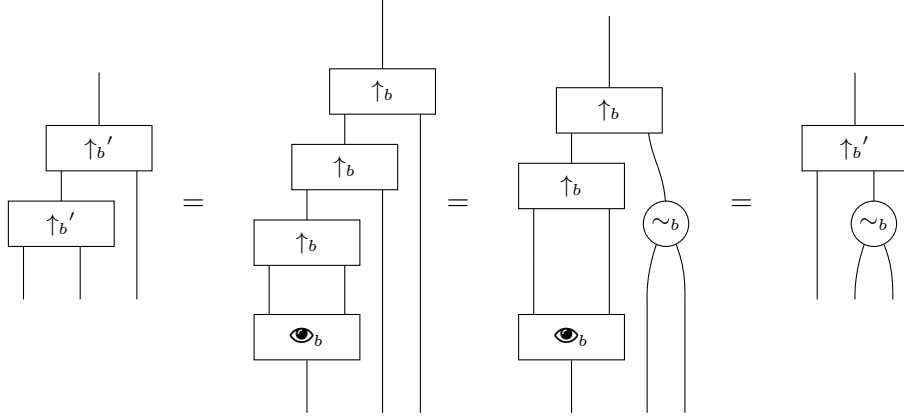
but the *repeat-update* axiom is easy to check. □

A.3 Proof of Proposition 3

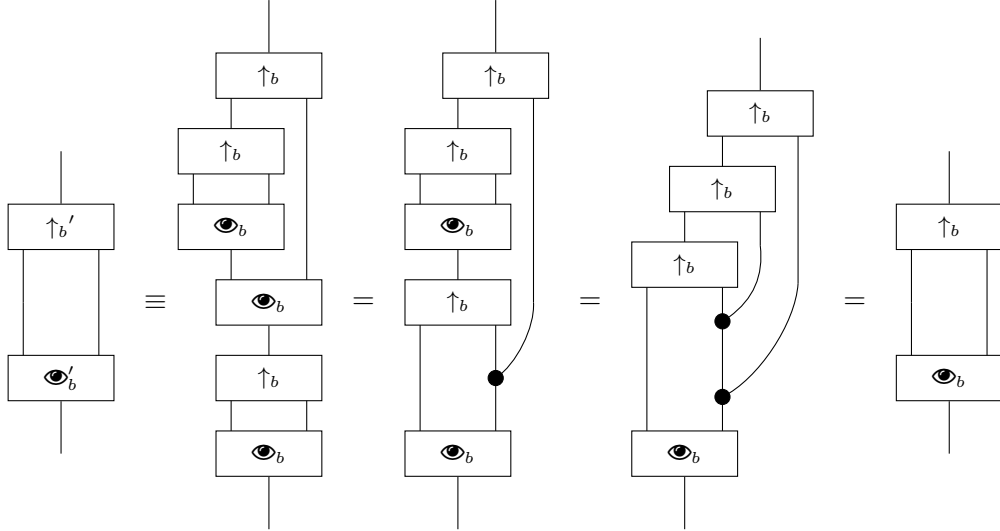
Making heavy use of the equation,



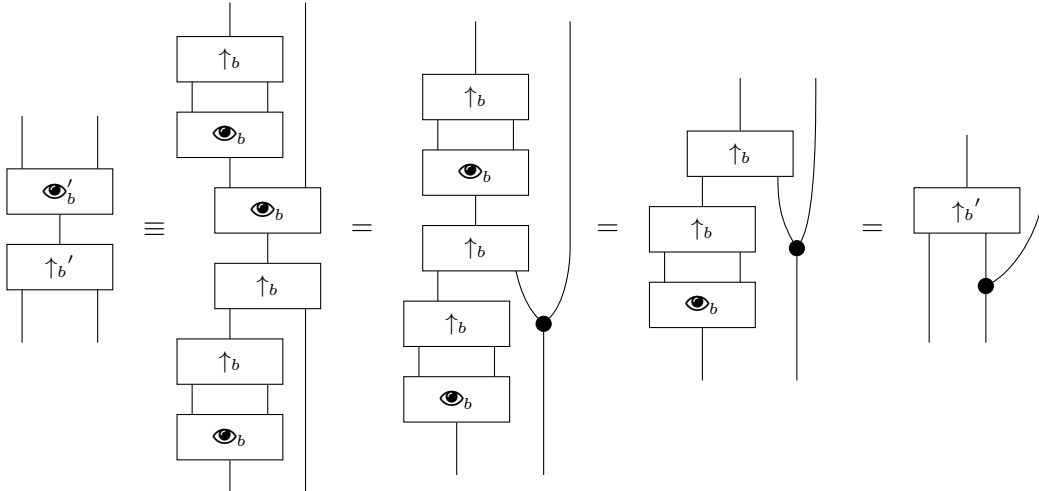
along with idempotency of $\uparrow_b \circ \text{eye}_b$ throughout, we begin by verifying PutPut,



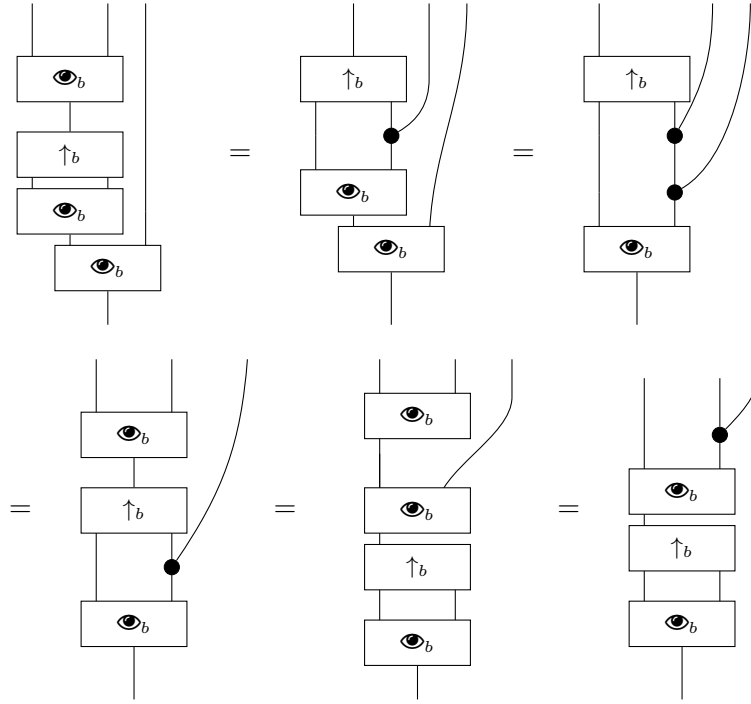
GetPut



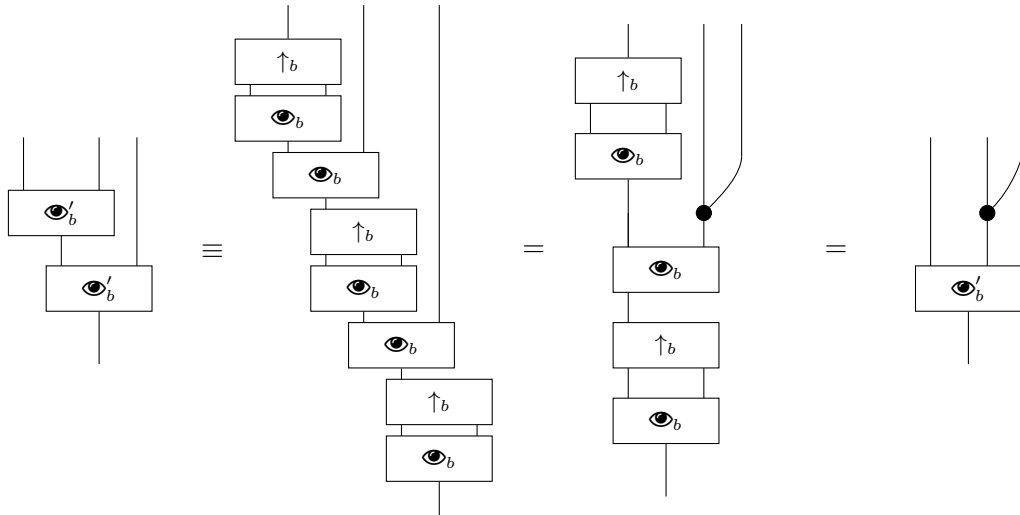
PutGet



Finally, since



GetGet follows



References

- [1] Samson Abramsky and Nikos Tzevelekos. Introduction to Categories and Categorical Logic. *arXiv:1102.1313 [cs, math]*, 813:3–94, 2010. arXiv: 1102.1313.
- [2] Yaared Al-Mehairi, Bob Coecke, and Martha Lewis. Compositional Distributional Cognition. August 2016.
- [3] F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Transactions on Database Systems*, 6(4):557–575, December 1981.
- [4] Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. Interacting Conceptual Spaces I : Grammatical Composition of Concepts. *arXiv:1703.08314 [cs]*, March 2017. arXiv: 1703.08314.
- [5] Bob Coecke. The Mathematics of Text Structure. *arXiv:1904.03478 [quant-ph]*, April 2019. arXiv: 1904.03478.
- [6] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 1 edition, 2017.
- [7] Bob Coecke and Dusko Pavlovic. Quantum measurements without sums. In G. Chen, L. Kauffman, and S. Lamonaco, editors, *Mathematics of Quantum Computing and Technology*, pages 567–604. Taylor and Francis, 2007. arXiv:quant-ph/0608035.
- [8] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical Foundations for a Compositional Distributional Model of Meaning. *arXiv:1003.4394 [cs, math]*, March 2010. arXiv: 1003.4394.
- [9] Bob Coecke, John Selby, and Sean Tull. Two roads to classicality. *EPTCS*, 266:104118, 2018.
- [10] J Nathan Foster, Bell Labs, Jonathan T Moore, and Alan Schmitt. Combinators for Bi-Directional Tree Transformations: A Linguistic Approach to the View Update Problem. *ACM Transactions on Programming Languages and Systems*, page 64.
- [11] Peter Grdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge, Mass., new ed edition edition, March 2004.
- [12] Chris Heunen, Aleks Kissinger, and Peter Selinger. Completely positive projections and biproducts. *EPTCS*, 171:7183, 2014.
- [13] Peter Selinger. Idempotents in dagger categories: (extended abstract). *Electronic Notes in Theoretical Computer Science*, 210:107 – 122, 2008. Proceedings of the 4th International Workshop on Quantum Programming Languages (QPL 2006).
- [14] Peter Selinger. A survey of graphical languages for monoidal categories. *arXiv:0908.3347 [math]*, 813:289–355, 2010. arXiv: 0908.3347.