STRING DIAGRAMS FOR TEXT

VINCENT WANG-MAŚCIANICA

ST. CATHERINE'S COLLEGE
THE UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
2023

Contents

(Acknowledgements will go in a margin note here.)

1

Sketches in iconic semantics

How to reason formally with and about pictorial iconic representations as a semantics of natural language.

1.1 Preliminary concepts for the sketches

This section should be read as a smooth transition from the contents of the previous chapter towards sketches that gradually trade off rigour for expressivity, while ideally being descriptive enough that the reader trusts that the necessary details can be worked out.

1.1.1 Open sets: concepts

Apart from enabling us to paint pictures with words, **ContRel** is worth the trouble because the opens of topological spaces crudely model how we talk about concepts, and the points of a topological space crudely model instances of concepts. We consider these open-set tests to correspond to "concepts", such as redness or quickness of motion. Figure 1.1 generalises to a sketch argument that insofar as we conceive of concepts in (possibly abstractly) spatial terms, the meanings of words are modellable as shared strategies for spatial deixis; absolute precision is communicatively impossible, and the next best thing mathematically requires topology.

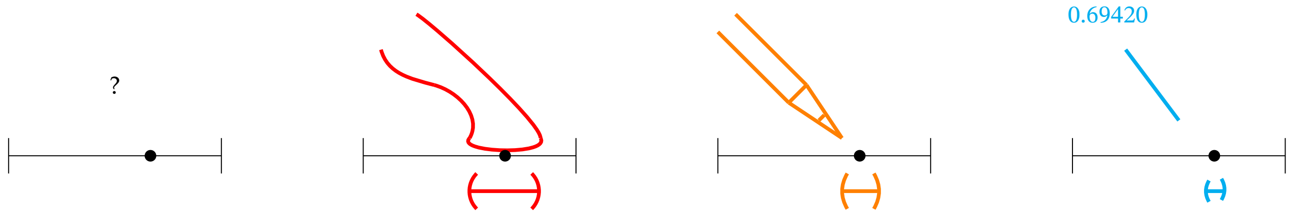
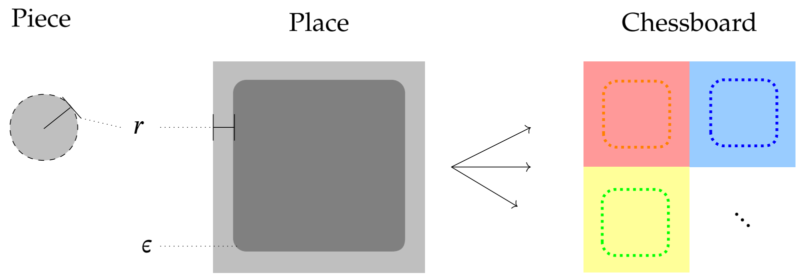


Figure 1.1: Points in space are a useful mathematical fiction. Suppose we have a point on a unit interval. Consider how we might tell someone else about where this point is. We could point at it with a pudgy appendage, or the tip of a pencil, or give some finite decimal approximation. But in each case we are only speaking of a vicinity, a neighbourhood, an *open set in the borel basis of the reals* that contains the point. Identifying a true point on a real line requires an infinite intersection of open balls of decreasing radius; an infinite process of pointing again and again, which nobody has the time to do. In the same way, most language outside of mathematics is only capable of offering successively finer, finite approximations.

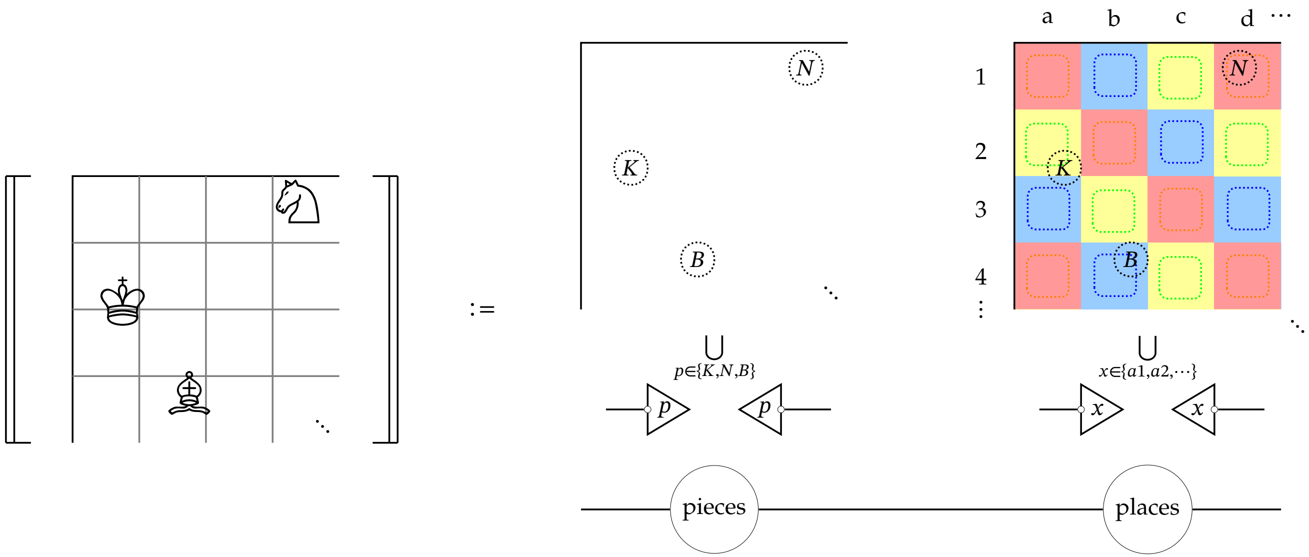
Maybe this explains the asymmetry of why tests are open sets, but why are states allowed to be arbitrary subsets? One could argue that states in this model represent what is conceived or perceived. Suppose we have an analog photograph whether in hand or in mind, and we want to remark on a particular shade of red in some uniform patch of the photograph. As in the case of pointing out a point on the real interval, we have successively finer approximations with a vocabulary of concepts: "red", "burgundy", "hex code #800021"... but never the point in colourspace itself. If someone takes our linguistic description of the colour and tries to reproduce it, they will be off in a manner that we can in principle detect, cognize, and correct: "make it a little darker" or "add a little blue to it". That is to say, there are in principle differences in mind that we cannot distinguish linguistically in a finite manner; we would have to continue the process of "even darker" and "add a bit less blue than last time" forever. All this is just the mathematical accommodation of a common observation: sometimes you cannot do an experience justice with words, and you eventually give up with "I guess you just had to be there". Yet the experience is there and we can perform linguistic operations on it.

1.1.2 Using sticky spiders as location-tests

Example 1.1.1 (Where is a piece on a chessboard?). How is it that we quotient away the continuous structure of positions on a chessboard to locate pieces among a discrete set of squares? Evidently shifting a piece a little off the centre of a square doesn't change the state of the game, and this resistance to small perturbations suggests that a topological model is appropriate. We construct two spiders, one for pieces, and one for places on the chessboard. For the spider that represents the position of pieces, we open balls of some radius r, and we consider the places spider to consist of square halos (which tile the chessboard), containing a core inset by the same radius r; in this way, any piece can only overlap at most one square.

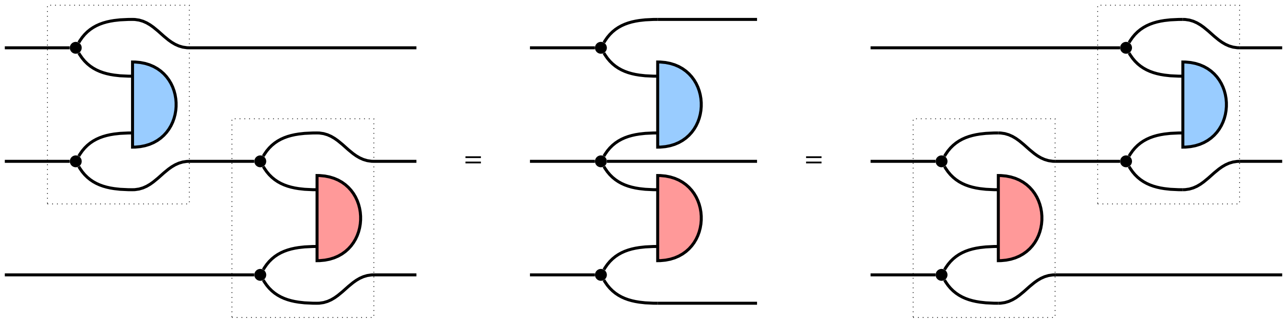


Now we observe that the calculation of positions corresponds to composing sticky spiders. We take the initial state to be the sticky spider that assigns a ball of radius r on the board for each piece. We can then obtain the set of positions of each piece by composing with the places spider. The composite (pieces;places) will send the king to a2, the bishop to b4, and the knight to d1, i.e. $\langle K| \mapsto \langle a2|, \langle B| \mapsto \langle b4|$ and $\langle N| \mapsto \langle d1|$. In other words, we have obtained a process that models how we pass from continuous states-of-affairs on a physical chessboard to an abstract and discrete game-state.



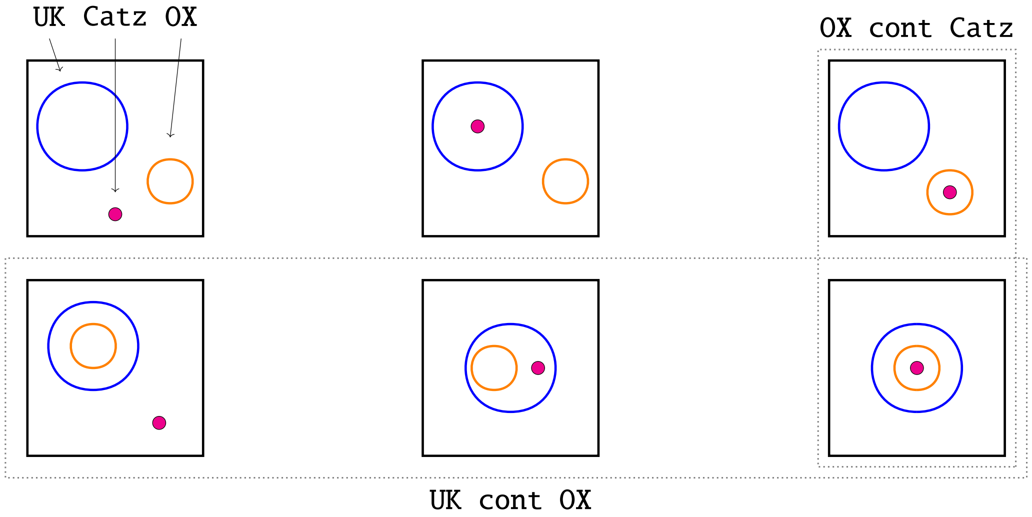
1.1.3 Copy: stative verbs and adjectives

Stative verbs are those that posit an unchanging state of affairs, such as `Bob likes drinking`. Insofar as stative verbs are restrictions of all possible configurations to a permissible subset, they are conceptually similar to adjectives, such as `red car`, which restricts permissible representations in colourspace. By interpreting conceptual spaces topologically, where concepts are particular open sets, we can test for whether states lie within concepts in the same way we can test whether a chesspiece is on a certain square. Moreover, **ContRel** conspires in our favour by giving us free copy maps on every wire, which allows us to define a family of processes that behave like stative restrictions of possibilities. These model stative verbs and adjectives.



The desirable property we obtain is that in the absence of *dynamic* verbs that posit a change in the state of affairs, stative constructions commute in text: if I'm just telling you static properties of the way things are, it doesn't matter in what order I tell you the facts because restrictions commute. Recall that gates of the following form are intersections with respect to open sets, and they commute. These intersections model conjunctive specifications of properties.

Example 1.1.2 (Containment and insideness)**.**



Consider the configuration space of a sticky spider on the unit square with three labelled shapes, which has 6 connected components, depicted. `Oxford contains Catz.` restricts away configurations where `Catz` is not enclosed in `Oxford`. Adding on `England contains Oxford.` further restricts away incongruent configurations, leaving us only with a single connected component, which contains all spatial configurations that satisfy the text. A similar story holds for abstract conceptual spaces, in which `fast red car`, `fast car that is red`, `car is (red and fast)` all mean the same thing.

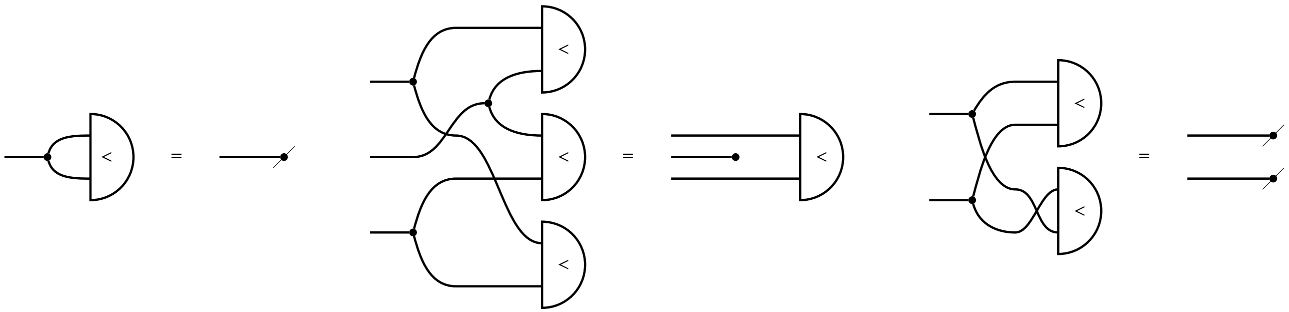
1.1.4 *The unit interval*

To begin modelling more complex concepts, we first need to extend our topological tools. Throughout, we now consider string-diagrams to be expressions that may be quantified over, and we allow ourselves additional niceties like endocombinators. Ultimately we would like to get at the unit interval so we can do homotopies to move shapes around, which we plan to arrive at by first expressing the reals, and then adding in endpoints. However, there are many spaces homeomorphic to the real line. How do we know when we have one of them? The following theorem provides an answer:

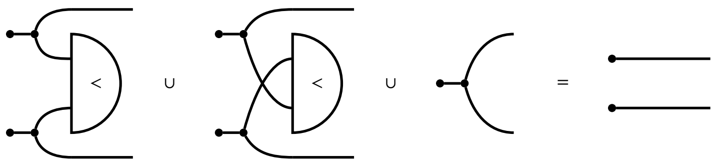
Theorem 1.1.3 ([Fri05])**.** Let $((X, \tau), <)$ be a topological space with a total order. If there exists a continuous map $f : X \times X \to X$ such that $\forall a, b_{\in X} : a < f(a, b) < b$, then X is homeomorphic to $\mathbb{R}$.

Definition 1.1.4 (Less than)**.** We define a total ordering relation $<$ as an open set on $X \times X$ that obeys the usual axiomatic rules:

Antireflexive ($\forall x : x \not< x$) Transitive ($\forall xyz : x < y \,\&\, y < z \Rightarrow x < z$) Antireflexive $\forall xy \neg(x < y \,\&\, y < x)$



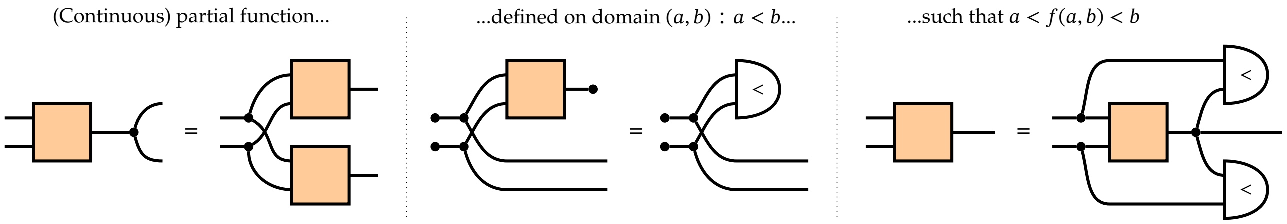
Trichotomy ($\forall xy : x < y \vee y < x \vee \; x = y$)



Definition 1.1.5 (Friedman's function)**.** Just as a wire in **ContRel** has the discrete topology if it possesses spider structure (Proposition **??**), a wire is homeomorphic to the real line by Theorem 1.1.3 if it possesses an

Postscript: If you're already happy that in principle we may either start with nicer spaces or otherwise restrict ourselves to contractible opens, then you may skip the next two subsections and just glance at how relational homotopies differ from regular homotopies, and look briefly at the definition of a nice spider at the end. The relevant conceptual takeaway for the couple of sketches is that one may recover the usual topological notions such as simple connectivity, metrics and their open balls, and contractibility, from which one can in principle construct models of linguistic topological relations such as `touching`, `enclosure`, and so on. The submitted version of this thesis had detailed constructions of these linguistic topological relations "from scratch" but I've cut them, so only the next two sketches remain as artifacts to suggest that "low-level" hacking in **ContRel** is doable. I've opted to remove sketches of linguistic topological relations because (1) they took up too much space for too little gain (2) they still admitted counterexamples, and (3) it seems plausible that any analysis of linguistic topological primitives in mathematical terms will admit counterexamples, because I suspect they have the status of semantic primes [Wie96], which are characterised by their universality across languages and their unanalysability in simpler terms.

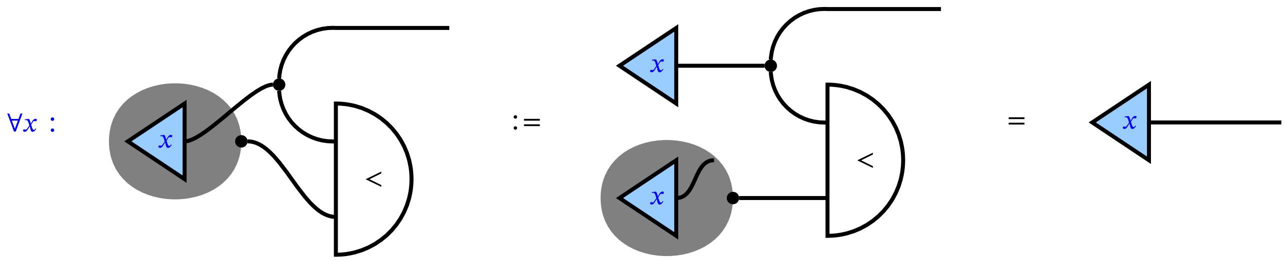
open that behaves as Definition 1.1.4, and a map that satisfies:



Let's say that the unit interval is like the real line extended with endpoints. One way to define this that aligns with the usual presentation of the reals in analysis is to provide the ability to take suprema and infima of subsets, which are functions that map subsets to points. This kind of function is subsumed by a kind of structure on a category called an endocombinator.

Definition 1.1.6 (Endocombinator)**.** An *endocombinator* on a category $\mathcal{C}$ is a family of functions on homsets typed $\mathcal{C}(X, Y) \to \mathcal{C}(X, Y)$, for all objects X, Y.

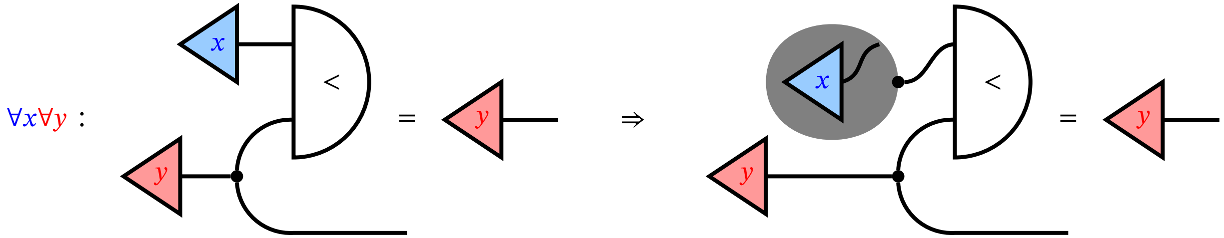
Definition 1.1.7 (Upper and lower bounds via endocombinators)**.** Upper bounds are endocombinators that send states to points, which we depict as a little gray lassoed region around the state of interest. Recall that points are states with a little decorating copy-dot as they are copiable. The following equational condition quantified over all states characterises an "upper bound" endocombinator that returns an upper bound for any subset of a totally ordered space: in prose, such subsets are all less than their upper bound.



We can add in further equations governing the upper bound endocombinator to turn it into a supremum, or least-upper-bound.

Definition 1.1.8 (Suprema)**.** An upper bound endocombinator is the supremum when the following additional condition (with caveats, see sidenote) holds: for all subsets y whose elements are all greater than those of a subset x, the supremum of x is less than all elements of y.

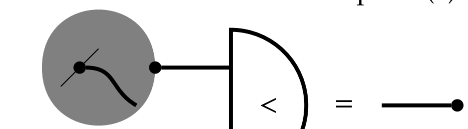
Unless y already contains $\sup(x)$, so the consequent of the implication needs a disjunctive case where $\sup(x) \cup y|_{\sup(x)<} = y$. The reason we cannot use $\leq$ as an open (even though it would make this definition easier) is that it would imply the equality relation $=$ is an open, which would imply that the underlying space has the discrete topology, trivialising everything.



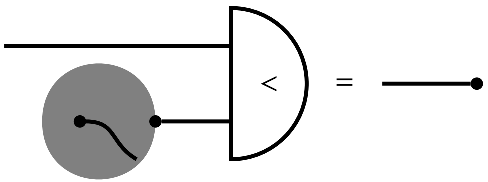
Now the lower endpoint is expressible as the supremum of the empty set, and the upper endpoint is the supremum of the whole set.

Definition 1.1.9 (Endpoints)**.** The lower endpoint is the supremum of the empty state, and the upper the supremum of everything.

Lower endpoint (0)



Upper endpoint (1)

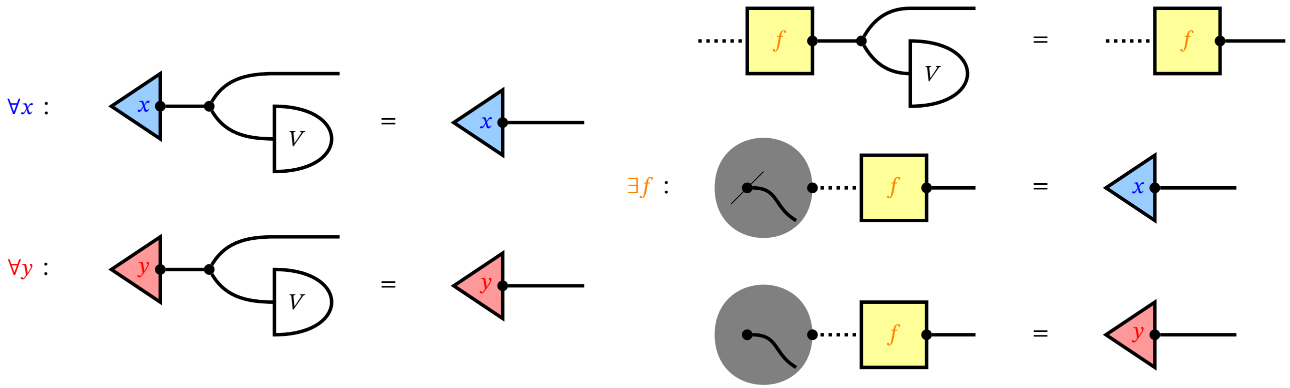


Definition 1.1.10 (The unit interval)**.** In **ContRel**, an object equipped with a less-than relation (Definition 1.1.4), Friedman's function (Definition 1.1.5), and suprema (Definitions 1.1.7 and 1.1.8) is homeomorphic to the unit interval. Going forward, we will denote the unit interval using a thick dotted wire.

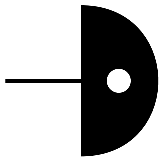
Conceptually, we are embedding the real line into a new space with two extra points, and then defining an extension of the less-than relation in terms of suprema to accommodate those points to characterise them as endpoints.

Example 1.1.11 (Simple connectivity)**.** Recall that we notate points and functions with the same small black dot for copying and deleting, as points are precisely the states that are copy-delete cohomomorphisms. In prose, simple connectivity states that for any pair of points that are within the open V, there exists some continuous function from the unit interval into the space that starts at one of the points and ends at the other. The left pair of conditions state that the points x and y are within V. The right triple of conditions require the the image of the homotopy f is contained in V, and that its endpoints are x and y.

V is *simply connected* when:

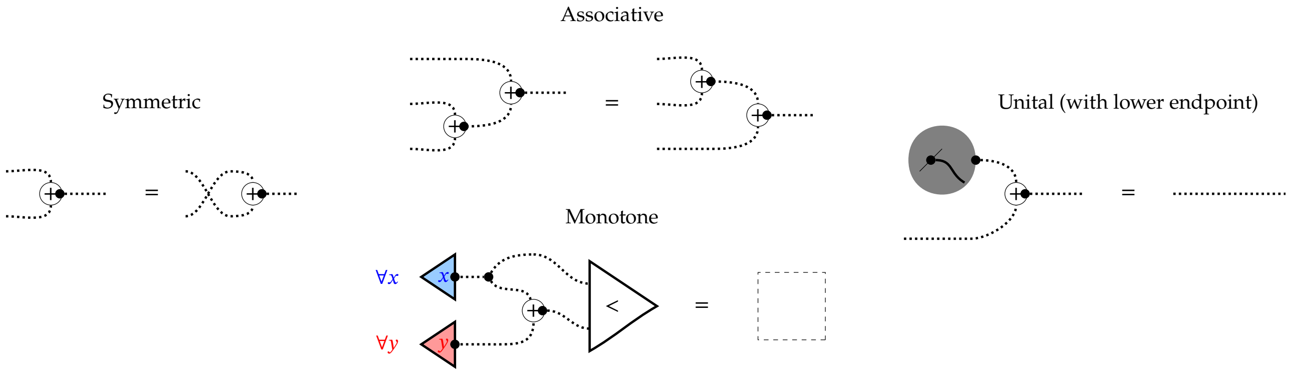


Simple connectivity is a useful enough concept that we will notate simply connected open sets as follows, where the hole is a reminder that simply connected spaces might still have holes in them.

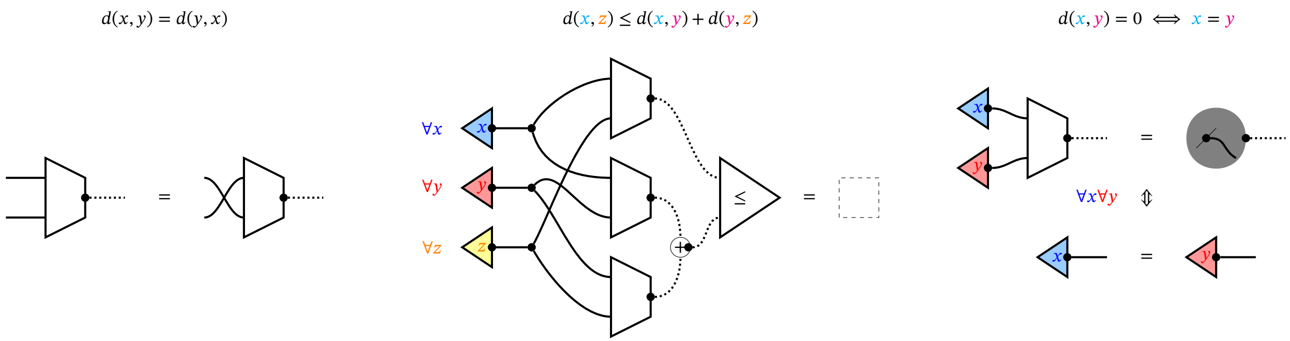


1.1.5 Metric structure

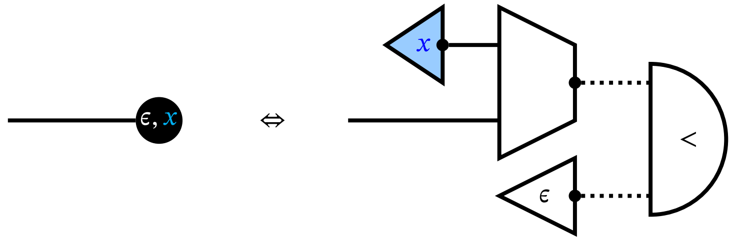
Definition 1.1.12 (Addition)**.** In order to define metrics, we must have additive structure, which we encode as an additive monoid that is a function. All we need to know is that the lower endpoint of the unit interval stands in for "zero distance" – as the unit of the monoid – and that adding positive distances together will deterministically give you a larger positive distance.



Definition 1.1.13 (Metric)**.** A metric on a space is a continuous map $X \to \mathbb{R}^+$ to the positive reals that satisfies the following axioms. We depict metrics as trapezoids.



Example 1.1.14 (Open balls)**.** Once we have metrics, we can define the usual topological notion of open balls. With respect to a metric, an ε-open ball at x is the open set (effect) of all points that are ε-close to x by the chosen metric.

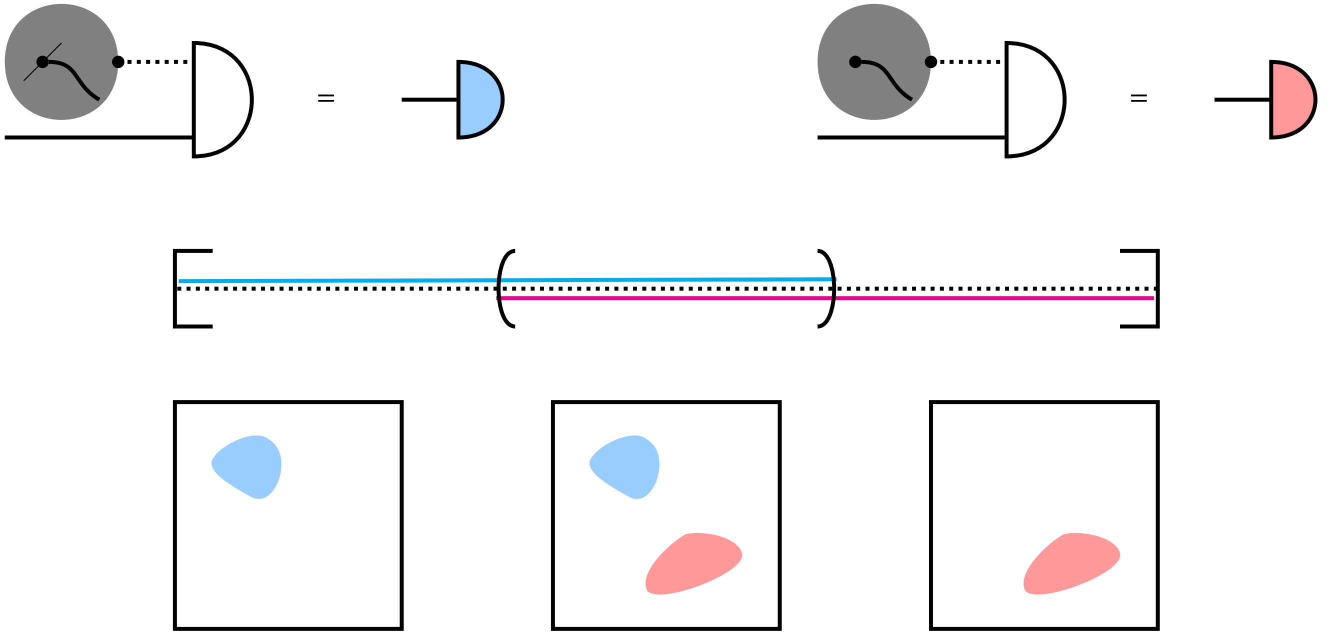


Open balls will come in handy later, and a side-effect which we note but do not explore is that open balls form a basis for any metric space, so in the future whenever we construct spaces that come with natural metrics, we can speak of their topology without any further work.

1.1.6 *Relational homotopy*

Definition 1.1.15 (Homotopy in **Top**)**.** where f and g are continuous maps $A \to B$, a *homotopy* $\eta : f \Rightarrow g$ is a continuous function $\eta : [0,1] \times A \to B$ such that $\eta(0,-) = f(-)$ and $\eta(1,-) = g(1,-)$.

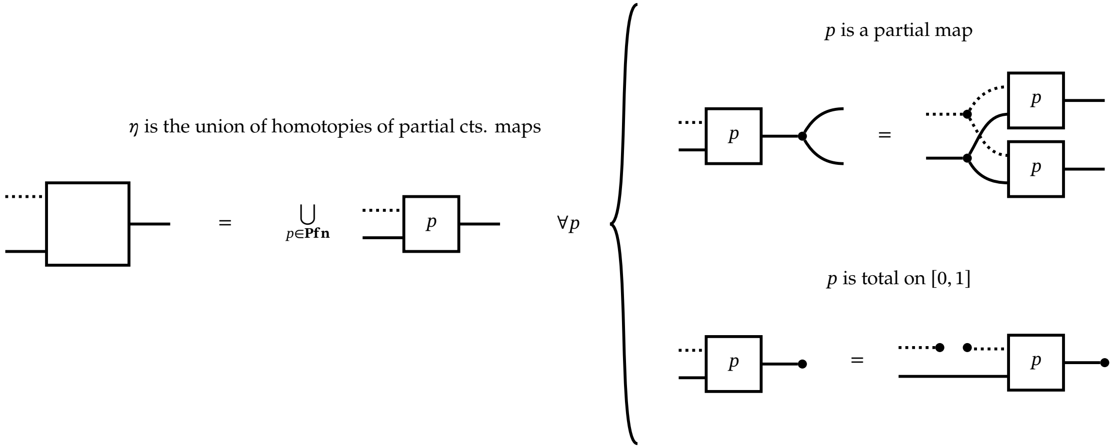
In other words, a homotopy is like a short film where at the beginning there is an f, which continuously deforms to end the film being a g. Directly replacing "function" with "relation" in the above definition does not quite do what we want, because we would be able to define the following "homotopy" between open sets.



What is happening in the above film is that we have a sticky spider expressing an open set in blue, which stays constant for a while. Then suddenly the ending open set in red appears (expressed by another sticky spider), and then the blue open disappears, and we are left with our ending; *technically* there was no discontinuity relative to the $[0,1]$-parameter in this relational homotopy between the two sticky spiders as endpoints, but there is something evidently discontinuous happening here that we would like to define away. The exemplified issue is that we can patch together (by union of continuous relations) vignettes of continuous relations that are not individually total on $[0,1]$. We can patch this issue by asking for relational homotopies in **ConRel** to satisfy the additional condition that they are expressible as a union of "partial homotopies" that are individually total on $[0,1]$.

Observe that the second condition asking for decomposition in terms of partial functions (of which total functions are a special case) comes for free by Proposition **??**, as the partial functions form a topological basis. the constraint of the definition is provided by the first condition, which is a stronger condition than just asking that the original continuous relation be total on I. Definition 1.1.16 is "natural" in light of Proposition **??**, that the partial continuous functions $A \to B$ form a basis for $\mathbf{ContRel}(A, B)$: we are just asking that homotopies between partial continuous functions – which can be viewed as regular homotopies with domain restricted to the subspace topology induced by an open set – form a basis for homotopies between continuous relations.

Definition 1.1.16 (Relational Homotopy)**.**



1.1.7 *Coclosure: adverbs and adpositions*

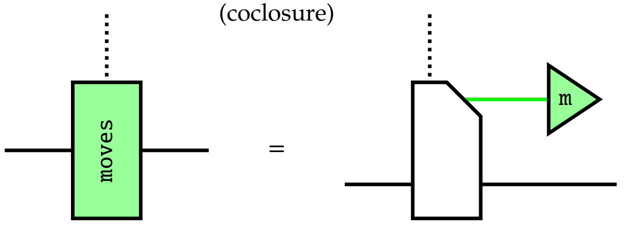


Figure 1.2: Recall that **ContRel** is coclosed (Proposition **??**), which means that every dynamic verb may be expressed as the composite of a coevaluator and an open set on the space of homotopies. For instance, `move` is an intransitive dynamic verb, which corresponds to a concept in the space of all movements.

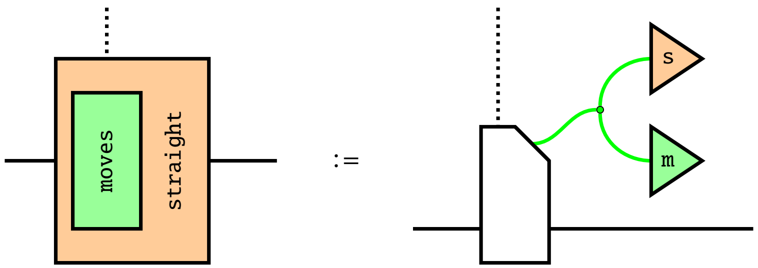


Figure 1.3: Adverb-boxes may be modelled as static restrictions in movement-space. For instance, `straight` may restrict movements to just those that satisfy some notion of path-length minimality: e.g., given a metric in movement-space on path-lengths, we may construct an open ball (Definition 1.1.14) around the geodesic to model the adverb `straight`.

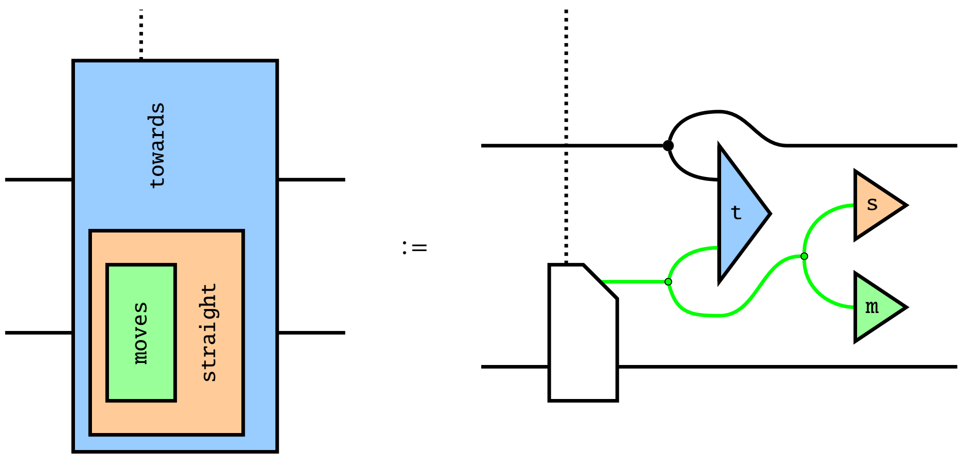
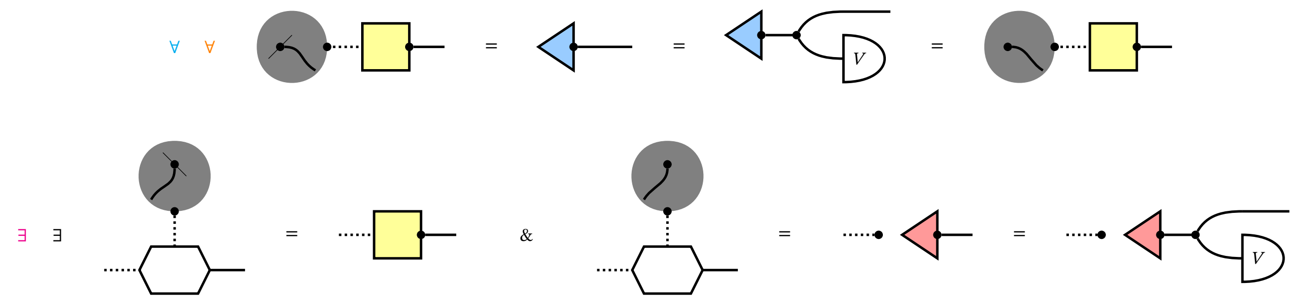


Figure 1.4: Similarly, adposition-boxes may be modelled as static restrictions on the product of the spaces of nouns and verbs. For instance, `towards` may be modelled as an open set that pairs potential positions of the thing-being-moved-towards with movements in movement-space that indeed move towards the target.

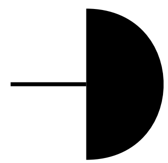
1.1.8 Nice spiders

Example 1.1.17 (Contractibility)**.** With homotopies in hand, we can define a stronger notion of connected shapes with no holes, which are usually called *contractible*. The reason for the terminology reflects the method by which we can guarantee a shape in flatland has no holes: when any loop in the shape is *contractible* to a point.

V is *contractible* when:

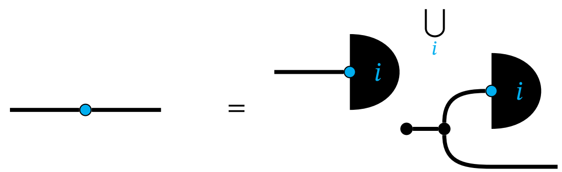


Contractible open sets are worth their own notation; a solid black effect, this time with no hole.



Let's assume for simplicity that henceforth, unless otherwise specified, we only deal with *nice* sticky-spiders where cores and halos agree and are both contractible opens; i.e. the spider can be expressed as a finite union of open solid blobs as effects followed by the same open solid blob as a state.

Definition 1.1.18 (Nice sticky-spiders)**.** A sticky-spider is *nice* if it is equal to a union of contractible open effects followed by the same contractible open expressed as a state.



1.2 Composition of dynamic verbs via temporal anaphora

Postscript: These sketches are mostly a restructuring of content that otherwise dangled from the previous chapter. Dynamic verbs and modals are two new sketches I had in mind while initially writing the thesis but didn't make it to the submitted version. There will probably be technical errors, but the sketches are not intended to be rigorous. None of these sketches (and nothing else in this thesis for that matter) should be taken as canonical once-and-for-all solutions to the conceptual problems they are meant to tackle; they are more meant to provoke as first-pass attempts, and they are meant to demonstrate how to play around and have fun in **ContRel** with string diagrams. I'll also note here that everything in **ContRel** is a kind of truth-conditional possible worlds semantics (up to some arbitrary but fixed choice of what particular ensembles of shapes and movements the modeller supplies up front), so there are no guarantees about how any of this material would fare if one tried to take the diagrams and interpret them in terms of neural networks, and I make no claims about whether the mathematics reflects actual cognition. However, I will claim that these mathematical sketches reflect at least the phenomenology of how *I* think about language, which should come as no surprise because my methodology was armchair introspection.

Dynamic verbs in iconic semantics may be modelled by homotopies, but non-parallel composition of homotopies is only defined up to parameters with indications of how the two separate homotopies begin and end relative to one another; i.e. temporal data.

Example 1.2.1 (Gluing homotopies sequentially at a time $\gamma \in (0,1)$)**.**

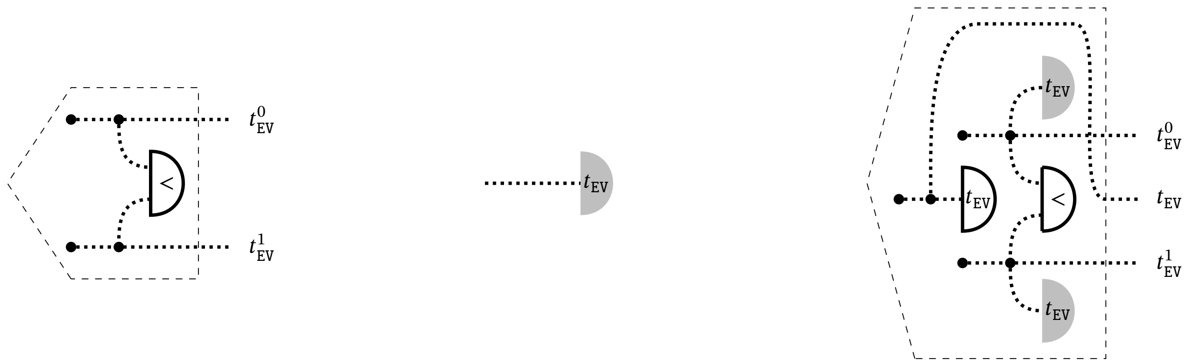
The technical difficulty I'd like to sketch a solution for is that while these parameters must be given as real numbers in the interval $[0,1]$, temporal natural language underspecifies: e.g. in the utterance `Bob drank, and then he slept` he could have drank in the morning and then slept in the afternoon, or both in the evening, and so on. The easy solution is to have absolute temporal anchors, but we seem to get by with less, which appears to necessitate a possible-worlds approach. Arguably the theoretical minimum we require is a kind of algebra for temporal aspects as in Yucatan [CITE], so here I sketch an algebra for temporal anaphora in **ContRel** that only requires copy-delete along with the standard topology on $\mathbb{R}$ obtained by the encoding of intervals as the open set $<: [0,1] \times [0,1]$. Then I'll show how this temporal data can be used to supply the information required for homotopy composition, which should indicate that **ContRel** is in-principle sufficiently expressive for dynamic iconic semantics for natural language, i.e. the interpretation of text as little moving cartoons.

Definition 1.2.2 (A sketch text-circuit algebra for temporal anaphora)**.** We consider three kinds of events. The first is episodic, which corresponds to some interval on $[0,1]$ with endpoints $t^0_{\texttt{EV}}$ and $t^1_{\texttt{EV}}$. We model these as bipartite states with the initial constraint that $t^0_{\texttt{EV}} < t^1_{\texttt{EV}}$. The second is habitual, which could in principle be an arbitrary subset of $[0,1]$, but there are pathologies we would like to rule out as a matter of common sense (e.g. we don't really talk about events that occur in time according cantor set), so we treat habituals as open sets (unions of intervals) to be later constructed or supplied as constraints; when we are finished specifying the algebra, equipping it with unions as a kind of formal sum will approximate those open sets that are constructible by finite amounts of talking about times. The third is a hybrid of the first two, where we consider some open set with distinguished endpoints, modelled as a restriction/intersection of an interval with some other open set.

Episodic event
(Interval determined by ordered endpoints)

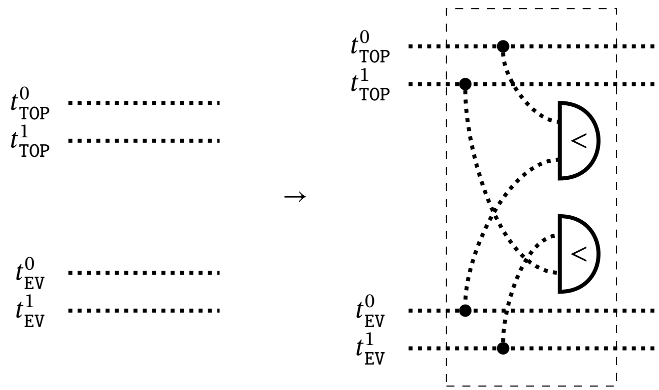
Habitual event (as constraint)
(An arbitrary open set on $[0,1]$)

Hybrid event
(Open set with endpoints)



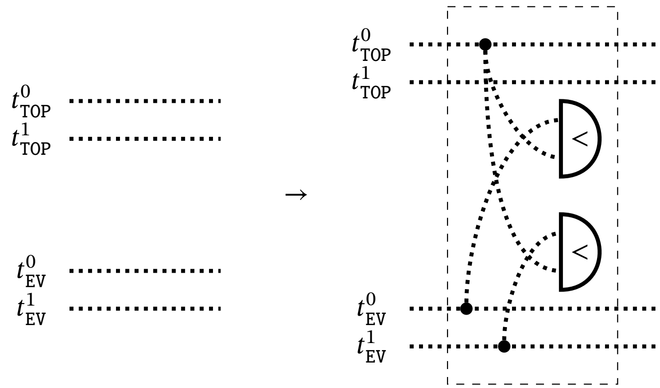
Now we model temporal aspects as circuit components — what appears to distinguish aspects from tenses is that aspects are always relative to the temporal data of two events, whereas tenses may be "intransitive" on events — so all of our aspectual data will involve constraining pairs of events (one of which is a `TOPIC`). The first kind of aspect we consider is *perfective*, which constraints an event time to be within topic time; we model this as imposing a constraint that the endpoints of the event must lie within the interval specified by the endpoints of the topic. In discourse, introducing a perfective constraint corresponds to adding a gate.

Perfective: $t_{\mathtt{EV}} \subseteq t_{\mathtt{TOP}}$
(Event time contained within topic time)



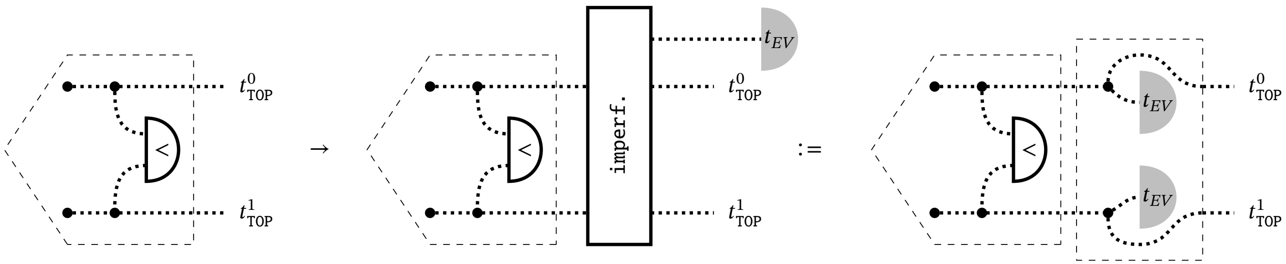
The *terminative* aspect constrains an event to occur entirely before the beginning of the topic time. Terminative composition of verbs may be glossed as `(event) and-then (topic)`, and this kind of composition yields the view of text circuits as implicitly encoding the temporal order in which gate-as-events occur, where now the sequential ordering of gates matters. This failure of interchange interprets text circuits in something like a premonoidal setting [CITE].

Terminative: $t_{\texttt{EV}} < t^0_{\texttt{TOP}}$
(Event will have been completed by the topic time)



The *imperfective* aspect we consider as constraining an episodic topic time to lie within some ongoing habitual event, where the habitual event is represented as a free coparameter. In discourse, introducing an imperfective constraint corresponds to splicing in such a constraint, which we gloss as a gate that restricts the endpoints of the topic interval to lie within the open set representing the habitual event time as a coparameter. We skip over the subtly distinct *progressive* aspect here as we won't need it for our later example, but it should be clear that an approach along these lines will also suffice.

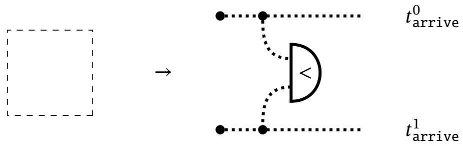
Imperfective: $t_{\texttt{TOP}} \subset t_{\texttt{EV}}$
(Episodic time contained within habitual event time)



Example 1.2.3. So here is an example of Yucatan Maya taken from [CITE], which is an excerpt of an interview

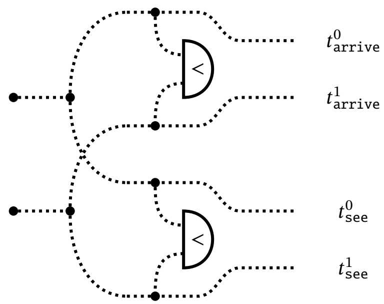
with a speaker fleeing a cyclone. I have split the excerpt into numbered single-verb clauses, accompanied by glosses in English with aspect-markers and the corresponding evolution of a text-circuit by the discourse rewrites we have defined. The first event introduced into discourse is the arrival of the refugees in the village, which is marked as perfective.

(1) [`Kaajk'ucho'on túun way tekàajil x Jaxleyile,` / `When we arrived,` (arrived: PERF.)] →



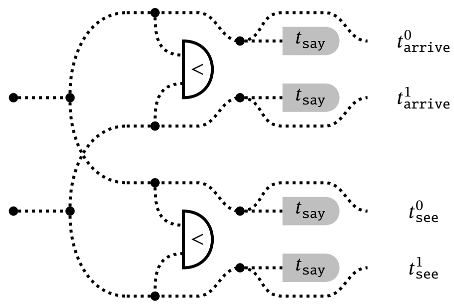
The second event is what the refugees saw, implicitly concurrent with event (1), which we opt to treat with a prepended copy of endpoints. `arrive & see` then form an atomic topic for events (3) and (4), which we deal with by constraining both (1) and (2) in the same way. Note that there is a single variable open set $t_{\texttt{say}}$ that is repeated 4 times in the diagram.

(2) [`kilike' tuláakal máake',` / `we saw,` (saw: PERF.)] →



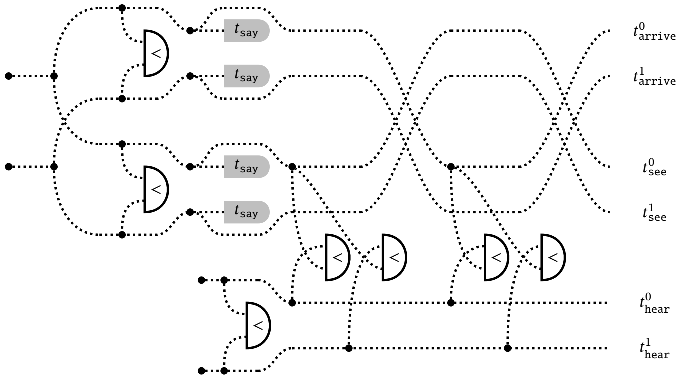
The third event refers to the villagers saying something, in the imperfective aspect with respect to events (1) and (2), so we constrain those topics accordingly. In gloss, it was an ongoing event that the villagers were saying something when the refugees arrived.

(3) [`táan uya'aliko'obe' jach` / `everyone was saying` (was saying: IMPF. WRT. (1,2))] →



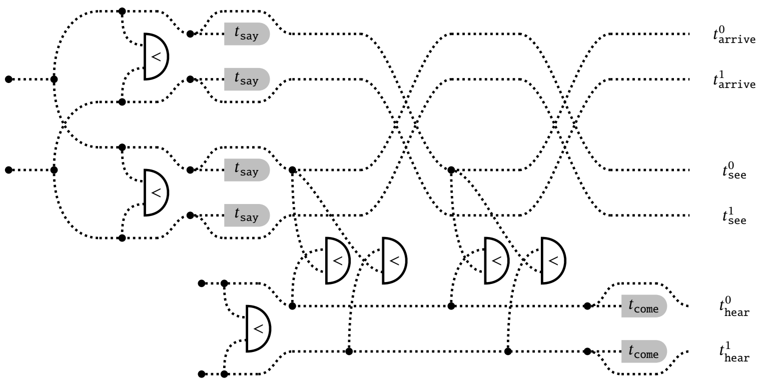
The fourth event refers to what the villagers had heard, in the terminative aspect with respect to (1) and (2). In gloss, the villagers were saying (reporting) the episodic event of them hearing something on the radio, and this hearing-event had completed before the refugees’ arrival.

(4) `ts’uyu’ubiko’ob ti’ ràadyoe’`
`(they) had heard on the radio` — *had heard*: TERM. WRT. (1,2) →



The fifth event refers the coming of the cyclone, which was ongoing at the time of the villagers hearing the radio report. This introduces a new habitual event as the variable open set $t_{\mathtt{come}}$, repeated twice in the diagram as constraints.

(5) `túun tàal le siklòono’.`
`the cyclone was coming.` — *was coming*: (4) IMPF. WRT. (5) →



Altogether, the final diagram represents a map from two open sets on $[0,1]$ (representing the potentially habitual events `say` and `come` encoded as variable open sets $t_{\mathtt{say}}$ and $t_{\mathtt{come}}$) to return a state in **ContRel** that encodes the set of possible endpoints for the episodic events `arrive`, `see` and `hear`: $\{(t^0_{\mathtt{arrive}}, t^1_{\mathtt{arrive}}, t^0_{\mathtt{see}}, t^1_{\mathtt{see}}, t^0_{\mathtt{hear}}, t^1_{\mathtt{hear}})\}$. Moreover, we have set up the algebra to allow us to leverage compositional discourse structure in such a way that sampling any of the elements of the resultant set returns a choice of endpoints consistent with the temporal constraints of the excerpt.

1.3 Iconic semantics for modal verbs

In this sketch I want to deal with certain modal verbs: that means those of cognition and perception like to think and see, and the sketch will taper out towards some modal auxiliaries like wanting. These kinds of verbs are roughly characterised as requiring copies of entities to be instantiated in worlds similar to but not exactly that of whatever base narrative reality is referred to in the discourse. For example, in `Alice sees Bob drink a beer, Bob drinks another after Alice leaves.`, there are two Bobs, because the one in Alice's mental-theatre drinks a single beer, and the one in the base reality of the narration drinks two. So there are two worlds $\mathfrak{W}$ here, one basic, and a $\mathfrak{W}_{\mathtt{A}}$ for the world in Alice's perception. Things get intractably tricky fairly quickly with these modals: to do epistemic logic means to have nested indices of what Alice thinks Bob thinks Alice thinks, to gossip is to reason about he-said-she-said, to understand complex narratives is to reason about stories-told-within-stories, and counterfactuals are a whole thing too. So that is a fundamental mystery: all this seems fairly complicated to encode and reason about symbolically, but it is phenomenologically fairly easy for adults to do, so what gives? What sort of mathematical presentation of these modals would at least reflect this lightness and ease?

I think thought-bubbles that show up in comic books are a pretty good start. Their cloudlike shape is a visual convention indicating a separate mental world, and they are typically used to represent want when the contents are also iconic representations.

Figure 1.5: Two examples by Mordillo, an artist I liked as a child: a thought bubble representing a woman, where the context of a stranded man implies a want for companionship, and a thought bubble representing a chair, where the context of a climber on a tall summit implies a want for rest.

The visual convention for cognitive and perceptive-alethic verbs is, as far as I can tell, a kind of x-ray effect into the contents of a head, which employs the familiar container metaphor: the head is a container for thoughts.

For alethic verbs in particular (those modals that are truth-preserving, in that they "do not forget" the truth), there's a need for the contents of the container to be synchronised with the contents of the outside world. Here are some observations that enable this in **Contrel**. The basic enabling insight is that, in Euclidean spaces, if we have a hollow container with a solid blob inside, there's an approximately continuous bijection between the (open set) insides of the container and the outside world.

Figure 1.6: On the left, a scene from the Simpsons showing the contents of Homer's mental-theatre. On the right, a depiction of two separate mental-theatres with a fisheye effect, taken from Steven Lahars "A Cartoon Epistemology" freely available online, which was also the initial inspiration for this sketch.

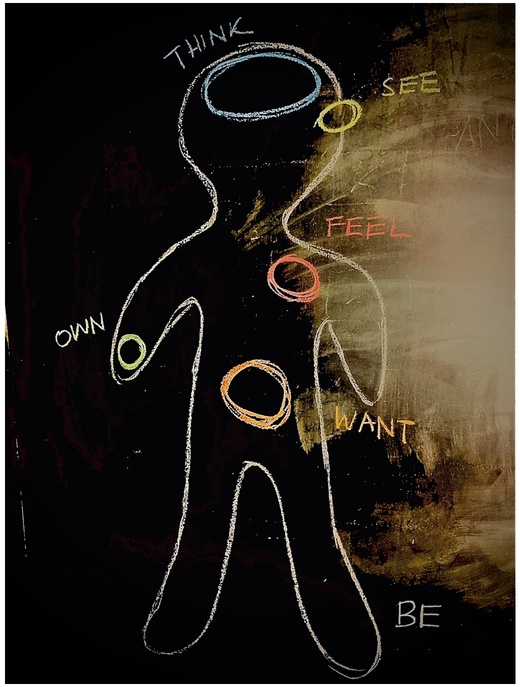


Figure 1.7: So the basic idea is to put representations of worlds inside bounded regions as containers, and in this way iconic semantics provides a univocal setting that displays all of the relevant worlds at once. We are free to pick visual conventions, as they are no more or less arbitrary than the assignment of indices and symbols such as $\mathfrak{W}_A$ to the contents of possible worlds. Here is a sketch convention for containers on an iconic representation of a person for different modal verbs: seeing, thinking, feeling, owning, and wanting. I sent this excitedly with little supporting context to Bob while I was writing my thesis. He was concerned. Then I got concerned. Childlike became creepy, and neither are good looks. I think I have supplied enough context to make this sensible, but there's no way I'm going to beat the crazy allegations.

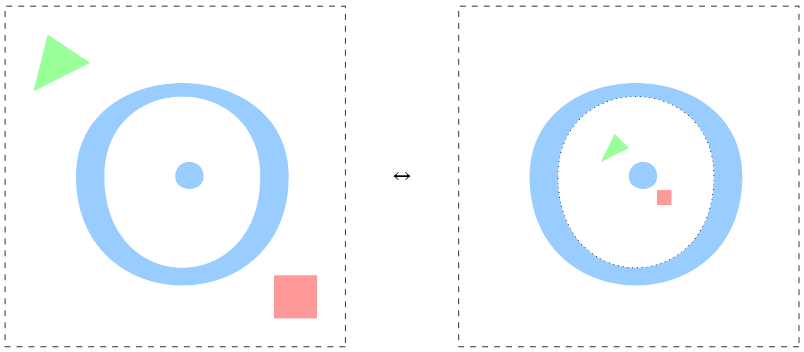


Figure 1.8: The inside and the outside of a container with a solid blob inside are both homotopic to the space with a puncture. This is only approximately a continuous bijection because the unbounded outside space can only map to the open interior of the container. We can use such bijections as a bridge to establish connections between elements of different possible worlds.

The second, and unfinished, idea is that if we have a handle on the individual components of sticky spiders, then we may use something like a very-well-behaved lens (hence its occurrence in the introduction) to ensure that the inside of the container is really behaving like a faithful storage medium for the goings-on outside. I think that's suggestive enough, and I'll deal with parthood in the next sketch. The last thing I want to deal with here is the problem of infinite regress for epistemic modals like knowing: if I know something, then I know I know it, and I know I know I know it, and so on. A naïve solution is to just use an infinitely-nested series of containers.

Figure 1.9: Again from Cartoon Epistemology, on the unsatisfactory nature of infinitely-nested containers: *But who is the viewer of this internal theatre of the mind? For whose benefit is this internal performance produced? Is it the little man at the center who sees this scene? But then how does HE see? Is there yet another smaller man inside that little man's head, and so on to an infinite regress of observers within observers?*

So the problem here is how to encode this infinite regress with finite means in an iconic model. The usual monadic approach still runs into the problem that you have to map a potential nested-infinity of possible worlds onto some finite model if one cares about cognitive realism. In iconic semantics, we can modify the space itself; here I think Escher was onto something.

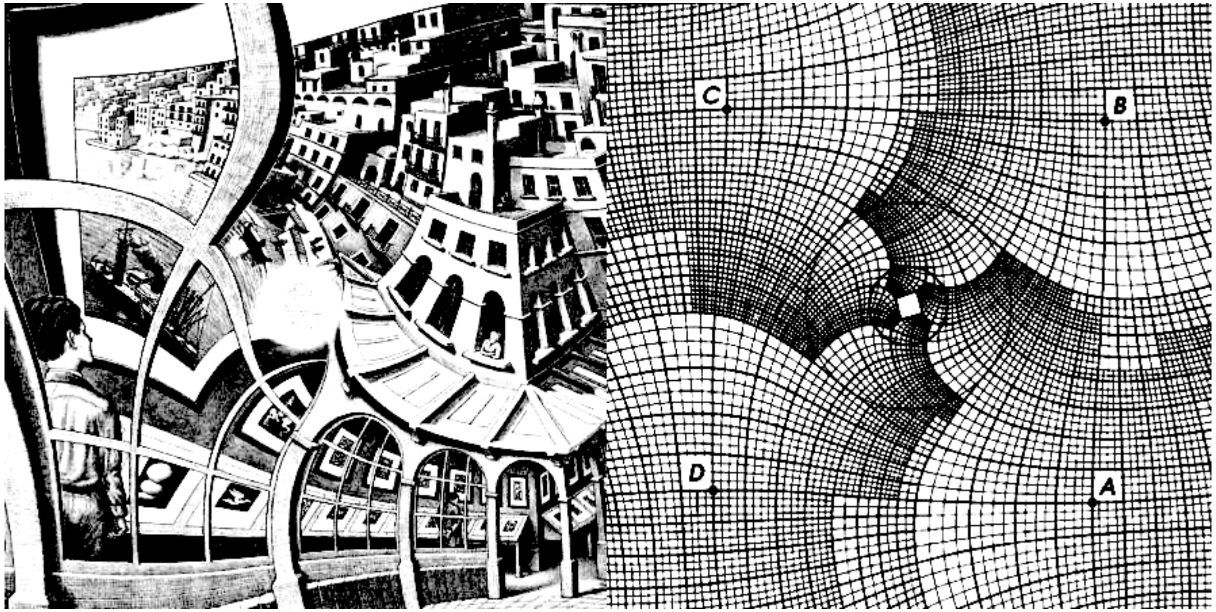


Figure 1.10: Escher's "Print Gallery" lithograph alongside his working sketch of the vortex-grid geometry the work was built on. On the left of the lithograph, an observer examines a framed painting of a town. Going clockwise, we see more details of the town, which has in it a print gallery, within which is the original observer. The missing centre of the piece where Escher signed the work obscures what would have been infinite nesting; the right-hand-side of the frame would have spiraled along the vortex infinitely. Treating the frame as a container, here we have an example of a container that contains itself, where movement clockwise indicates going down a level, clockwise going up, yet no explicit infinities anywhere.

The space in which such an arrangement can be realised is the same as that of the Penrose staircase: splitting the lithograph into four corners, each is a locally consistent snapshot, each gluing of quadrants is a consistent (as/de)scent, but the overall manifold obtained needs to be embedded in a higher dimension. While this in principle solves the problem of finitely representing infinite descent, these kinds of spaces are not grounded in physical, embodied intuitions. I think it is mathematically neat that there can exist topological models for such modal verbs, but whether such proposals are to be taken seriously as modelling cognition is a thorny matter I don't want to say more about.

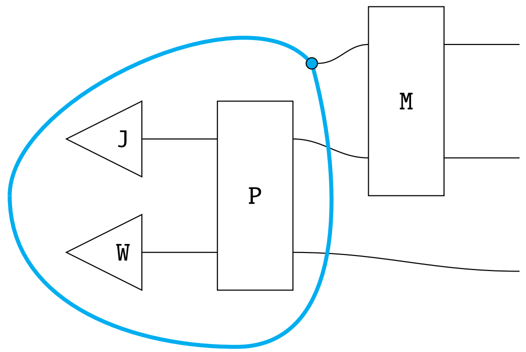
1.4 *Iconic semantics for general anaphora via Turing objects*

This sketch complements the sketch on modals, as it relies on the same container-trick. I would like to explain here how iconic semantics in **ContRel** might model untyped-boxes — these are conjunctions and verbs with sentential complements — as well as the more general linguistic phenomena of *entification* and *general anaphora* — where arbitrary discourse elements up to collections of sentences may be packaged up as if they were nouns and referred to. I suggest that the mathematical property of **ContRel** that enables this is that it contains **FinRel** equipped with a *Turing object*.

Entification is the process of turning words and phrases that aren't nouns into nouns. We are familiar with morphological operations in English, such as *inflections* that turn the singular `cat` into the plural `cats`, by adding a suffix `-s`. Another morphological operation generally called *derivation* changes the grammatical category of a word: for example, the adjective `happy` derives the noun `happiness`. With suffixes such as `-ness` and `-ing`, just about any lexical word in English can be turned into a noun, as if lexical words have some semantic content that is independent of the grammatical categories they might wear as a guise. With more complex discourse prefixes such as `the fact that`, we may also disguise sentences and text as nouns.

Example 1.4.1. Generalised anaphora as entification.

`Jono is paid minimum wage.  He didn't mind it.`

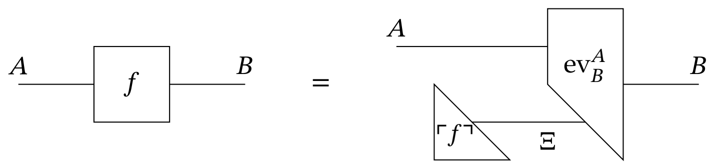


An example of entification. It may be argued that `it` refers to `the fact that Jono was paid minimum wage`. Graphically, we might want to depict the gloss as a circuit with a lasso that gives another noun-wire that encodes the information of the lassoed part of the circuit.

The problem at hand is finding an appropriate mathematical setting to interpret and calculate with such lassos. In principle, any meaningful (possibly composite) part of text can be referred to as if it were a noun. For syntax, this is a boon; having entification around means that there is no need to extend the system to accommodate wires for anything apart from nouns, so long as there is a gadget that can turn anything into a noun and back. For semantics this is a challenge, since this requires noun-wires to "have enough space in them" to accommodate full circuits operating on other noun-wires, which suggests a very structured sort of infinity. Computer science has had a perfectly serviceable model of this kind of noun-wire for a long time. What separates a computer from other kinds of machine is that a computer can do whatever any other kind of machine could do — modulo church-turing on computability and the domain of data manipulation — so long as the computer is running the right *program*. Programs are (for our purposes) processes that manipulate variously formatted — or typed — data, such as integers, sounds, and images. They can operate in sequence and in parallel, and wires can be swapped over each other, so programs form a process theory, where we can reason about the extensional equivalence of different programs — whether two programs behave the same with respect to mapping inputs to outputs. What makes computer programs special is that on real computers, they are specified by *code*. Programs that are equivalent in their extensional behavior may have many different implementations in code: for example, there are many sorting algorithms, though all of them map the same inputs to the same outputs. Conversely, every possible program in a process theory of programs must have some implementation as code. Importantly, code is just another format of data. The process-theoretic characterisation of the code-wire in a process-theory of computation is this:

Definition 1.4.2 (Turing object)**.** A *Turing object* Ξ in a process-theory is equipped with evaluation morphisms $\mathrm{ev}_B^A : A \otimes \Xi \to B$ for all pairs of objects A, B such that for all morphisms $f : A \to B$, there exists a state $\ulcorner f \urcorner_{: I \to \Xi}$ of the Turing object such that partial evaluation with that state is equal to f. The diagrammatic convention and visual pun [Pav23] for such code-states and evaluators is to depict the state-triangle as if it is cut out from the rectangle of the evaluator.

$$\forall A, B \in Ob(\mathcal{C})\ \exists \mathrm{ev}_B^A\ \forall f\ \exists \ulcorner f \urcorner$$



Any programming language is a model for text circuits, using the code-data format as the noun wire and Turing object. In **ContRel**, the unit square suffices as a Turing object for finite sets and relations, as we can use the container-trick of modals.

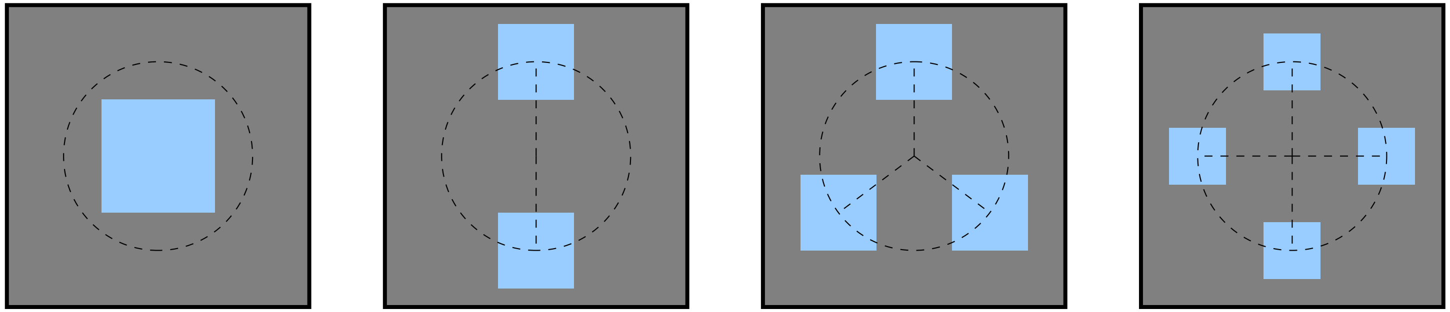
Proposition 1.4.3 (Sticky spiders on the open unit square model **FinRel** equipped with a Turing object)**.** Using the open unit square with its usual topology as the Turing object, there is a subcategory of **ContRel** which behaves as the category of countable sets and relations equipped with a Turing object

Proof. By Construction 1.4.9, which we work towards. □

Another observation we could have made is that since computers really just manipulate code, every data format is a kind of restricted form of the same Turing object Ξ, but this turns out to be a mathematical consequence of the above equation (and the presence of a few other operations such as copy and compare that form a variant of frobenius algebra), demonstrated in Pavlovic's forthcoming monoidal computer book [Pav23], itself a crystallisation of three monoidal computer papers [Pav12, Pav14, PY18]. I would be remiss to leave out Cockett's work on Turing categories [CH08], from which I took the name Turing object. Both approaches to a categorical formulation of computability theory share the common starting ground of a special form of closure (monoidal closure in the case of monoidal computer and exponentiation in Turing categories) where rather than having dependent exponential types $\mathbb{A} \multimap \mathbb{B}$ or $\mathbb{B}^{\mathbb{A}}$, there is a single "code-object" Ξ. They differ in the ambient setting; Pavlovic works in the generic symmetric monoidal category, and Cockett with cartesian restriction categories, which generalise partial functions. I work with Pavlovics' formalism because I prefer string diagrams to commuting diagrams.

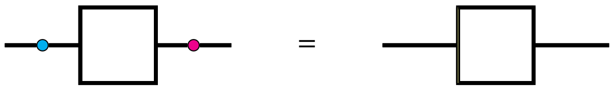
Lemma 1.4.4 ($(0,1) \times (0,1)$ splits through any countable set X)**.** For any countable set X, the open unit square ▦ has a sticky spider that splits through $X^{\star}$.

Proof. Proof by construction. Assume we work with nice spiders (.............), so we only have to highlight the copiable open sets. Take some circle and place axis-aligned open squares evenly along them, one for each element of X. The centres of the open squares lie on the circumference of the circle, and we may shrink each square as needed to fit all of them.

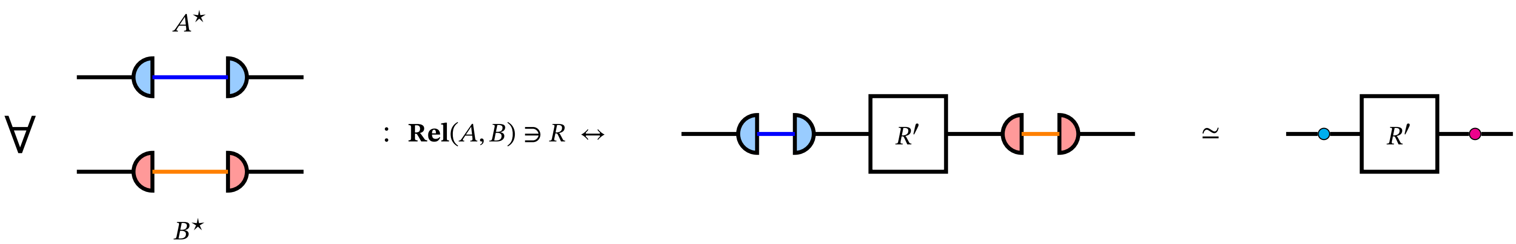


□

Definition 1.4.5 (Morphism of sticky spiders)**.** A morphism between sticky spiders is any morphism that satisfies the following equation.

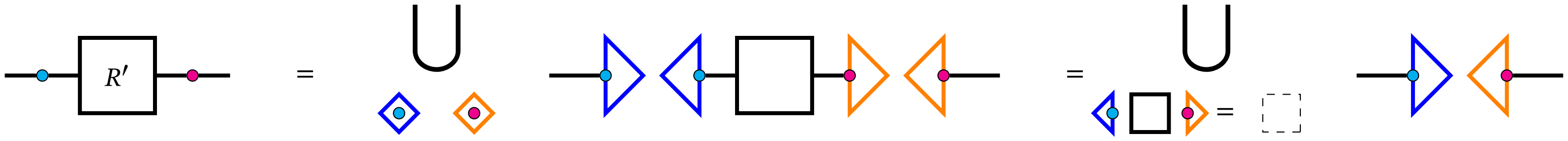


Lemma 1.4.6 (Morphisms of sticky spiders encode relations)**.** For arbitrary split idempotents through $A^{\star}$ and $B^{\star}$, the morphisms between the two resulting sticky spiders are in bijection with relations $R : A \to B$.



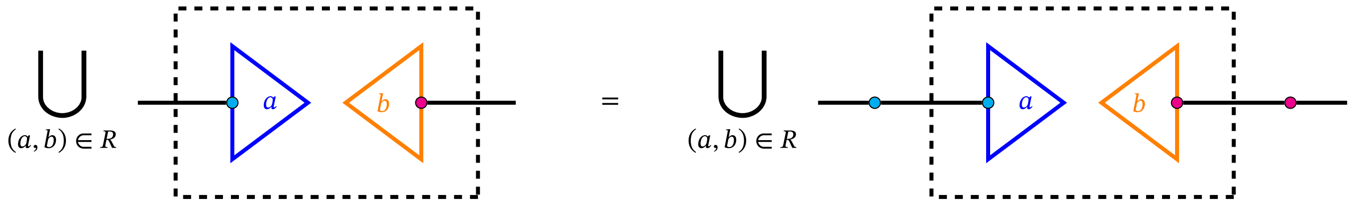
Proof.

(⇐) : Every morphism of sticky spiders corresponds to a relation between sets.



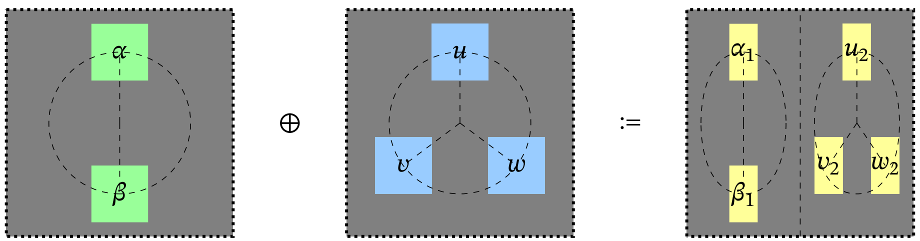
Since (co)copiables are distinct, we may uniquely reindex as:

(⇒) : By idempotence of (co)copiables, every relation $R \subseteq A \times B$ corresponds to a morphism of sticky spiders.

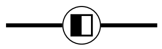


□

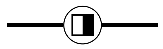
Construction 1.4.7 (Representing sets in their various guises within ▦). We can represent the direct sum of two ▦-representations of sets as follows.



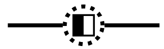
The important bit of technology is the homeomorphism that losslessly squishes the whole unit square into one half of the unit square. The decompressions are partial continuous functions, with domain restricted to the appropriate half of the unit square.



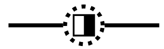
$(x, y) \mapsto (\frac{x}{2}, y)$



$(x, y) \mapsto (\frac{x+1}{2}, y)$



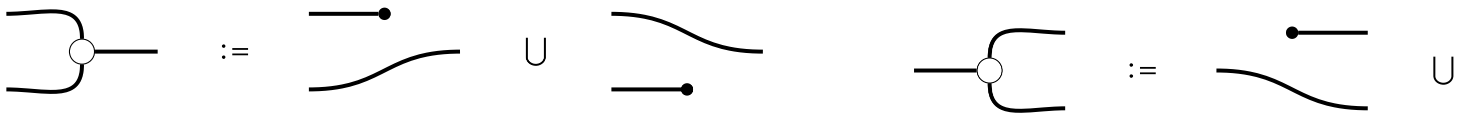
$(x, y)|_{x<\frac{1}{2}} \mapsto (2x, y)$



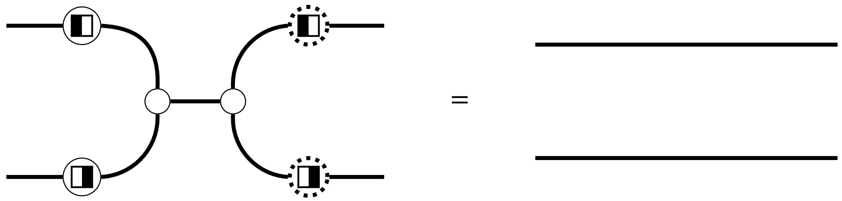
$(x, y)|_{x>\frac{1}{2}} \mapsto (2x - 1, y)$

We express the ability of these relations to encode and decode the unit square in just either half by the following graphical equations.

Now, to put the two halves together and to take them apart, we introduce the following two relations. In tandem with the squishing and stretching we have defined, these will behave just as the projections and injections for the direct-sum biproduct in **Rel**.

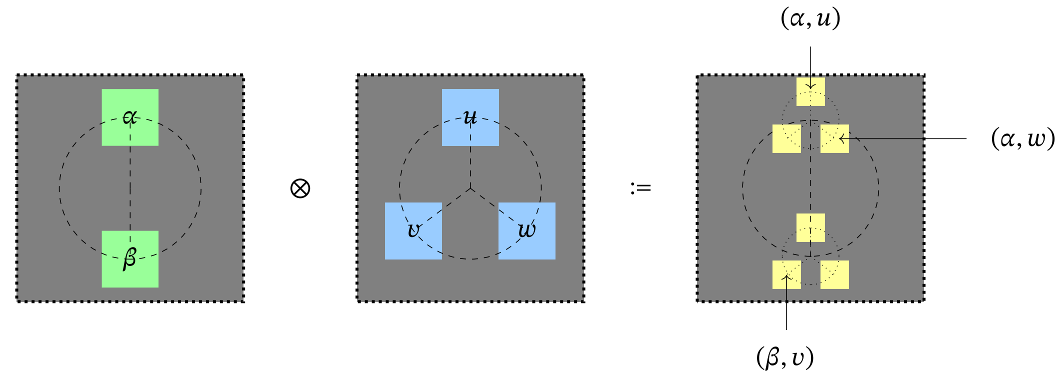


The following equation tells us that we can take any two representations in ▦, put them into a single copy of ▦, and take them out again.



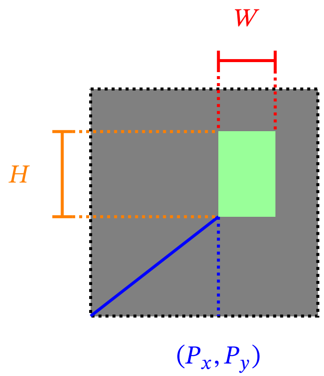
We encode the tensor product $A \otimes B$ of representations by placing copies of B in each of the open boxes of

A.



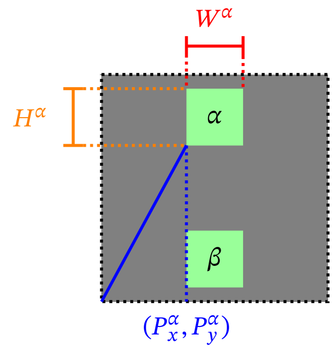
The important bit of technology here is a family of homeomorphisms of ▦ parameterised by axis-aligned open boxes, that allow us to squish and stretch spaces. Thus for every representation of a set in ▦ by a sticky-spider, where each element corresponds to an axis-aligned open box, we can associate each element with a squish-stretch homeomorphism via the parameters of the open box, which we notate with a dot above the name of the element.

$(x, y) \mapsto (Wx + P_x, Hy + P_y)$ $\quad (x, y)|_{\in(P_x, P_x+W)\times(P_y, P_y+H)} \mapsto (\frac{x-P_y}{W}, \frac{y-P_y}{H})$



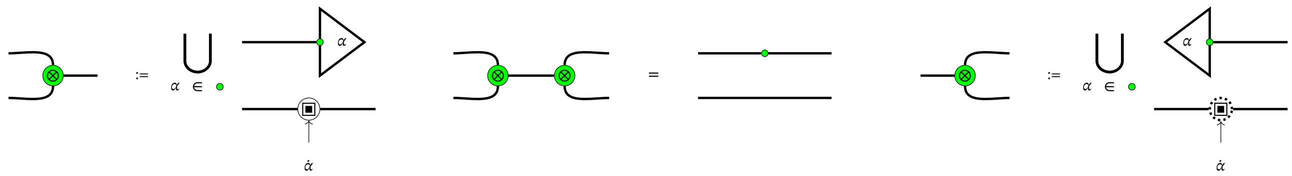
$\dot{\alpha} = (W^\alpha, H^\alpha, P_x^\alpha, P_y^\alpha)$

$\dot{\beta} = (W^\beta, H^\beta, P_x^\beta, P_y^\beta)$

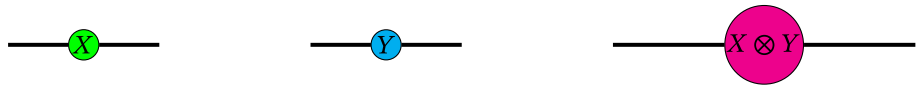


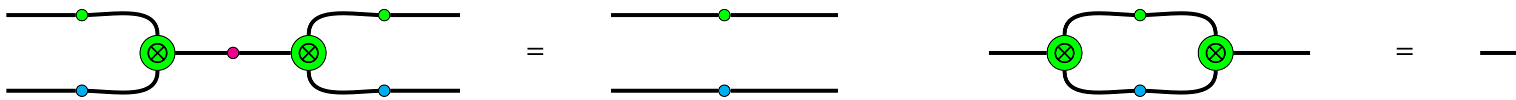
Remark 1.4.8. The essential idea is that the whole of the unit square is homeomorphic to part of it. In particular this means (modulo a point), we may make copies of shapes outside a container that are in homeomorphic correspondence with shapes within a container, the classic example being thought bubbles in a comic-strip with pictorial contents of the outside world. In our framework, this constitutes a formal and well-typed semantics for certain alethic verbs of cognition such as `sees` and `thinks`. For other such verbs, one may by Construction 1.4.9 operate directly on set-theoretic representations of mental contents.

Now we can define the "tensor X on the left" relation _ $\to X \otimes$ _ and its corresponding cotensor.

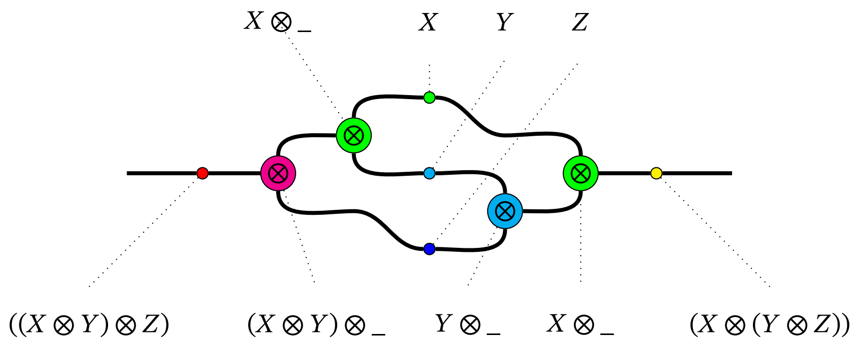


The tensor and cotensor behave as we expect from proof nets for monoidal categories.

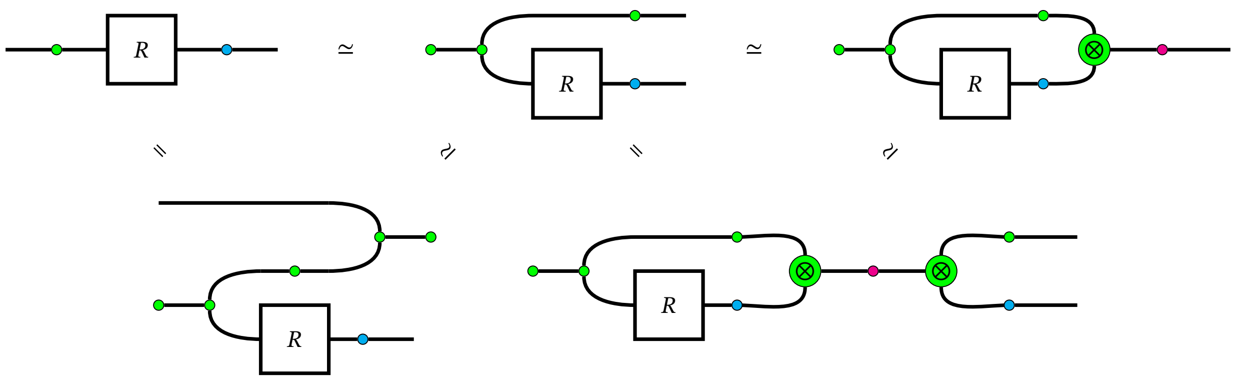




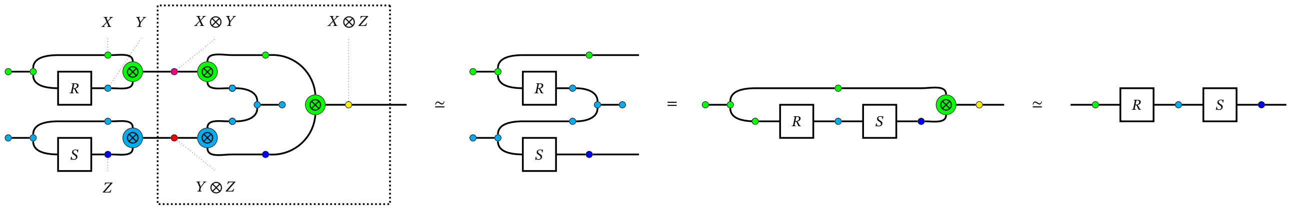
And by construction, the (co)tensors and (co)pluses interact as we expect, and they come with all the natural isomorphisms between representations we expect. For example, below we exhibit an explicit associator natural isomorphism between representations.



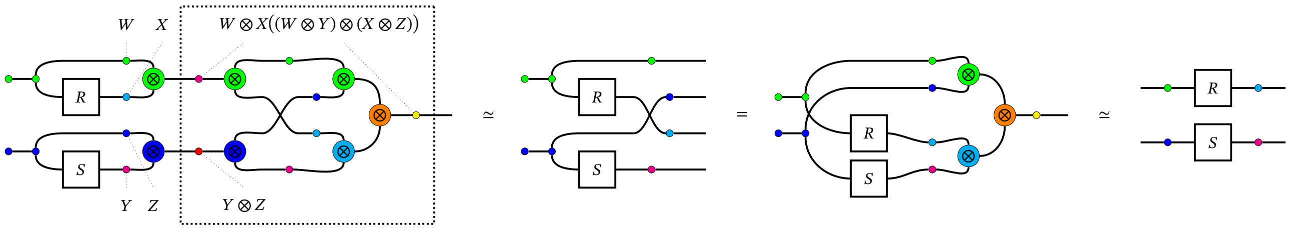
Construction 1.4.9 (Representing relations between sets and their composition within ▦)**.** With all the above, we can establish a special kind of process-state duality; relations as processes are isomorphic to states of ▦, up to the representation scheme we have chosen. This is part of the condition for Turing objects. What remains to be demonstrated is that the duality coheres with sequential and parallel relational composition.



Under this duality, we have continuous relations that perform sequental composition of relations as follows.



And similarly, parallel composition. Therefore, we have demonstrated that the unit square behaves as a Turing object for the category of countable sets and relations.



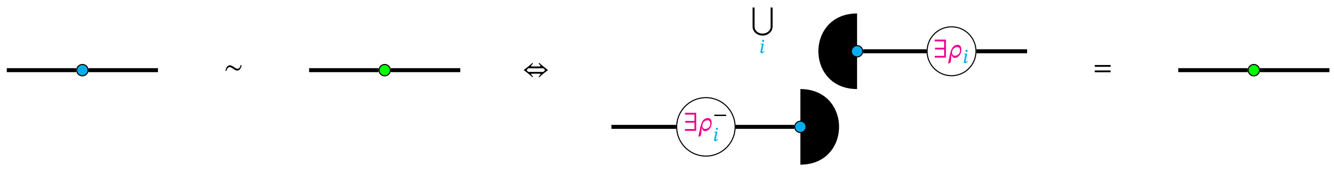
1.5 Configuration spaces

[1] [

[1]]Configuration spaces and the possible-worlds semantics they enable are a good reason to work string-diagrammatically in **ContRel** rather than **Top**, because the latter is cartesian monoidal, which in particular means that there is only one effect (the map into the terminal singleton topology), and worse, all states are tensor-separable. The latter fact means that we cannot reason natively in diagrams about correlated states, which are analogous to entangled quantum states [CK17], and spatial relations [WMC21]. I'll briefly explain here the gist of the analogy in prose because it is already presented formally in the cited works and elaborated in [Coe21]. The Fregean notion of compositionality is roughly that to know a composite system is equivalent to knowing all of its separate parts, and diagrammatically this amounts to tensor-separability, which arises as a consequence of cartesian monoidality. Schrödinger suggests an alternative of compositionality via a lesson from entangled states in quantum mechanics: *perfect knowledge of the whole does not entail perfect knowledge of the parts.* Let's say we have information about a composite system if we can a priori restrict the range of possible outcomes; this is the case for the bell-state, where we know that there is an even chance both qubits measure up or both measure down, and we can rule out mismatched measurements. However, discarding one entangled qubit from a bell-state means we only know that the remaining qubit has a 50/50 of measuring up or down, which is the minimal information (maximal entropy) we can have about a qubit. So we have at least one concrete example where we can know something about the whole, but nothing about its parts. A more familiar example from everyday life is if I ask you to imagine `a cup on a table in a room`. There are many ways to envision or realise this scenario in your mind's eye, all drawn from a restricted set of permissable positions of the cup and the table in some room. The spatial locations of the cup and table are entangled, in that you can only consider the positions of both together. If you discard either the cup or the table from your memory, there are no restrictions about where the other object could be in the room; that is, the meaning of the utterance is not localised in any of the parts, it resides in the entangled whole. Individual sticky-spiders correspond to static collections of set-labelled shapes in **ContRel**; in this sketch I want to talk about all the different ways the same collection of shapes can be arranged in space.

Let's also say we start with the ability to detect whether two sticky-spiders are related to one another by rigid displacements, expressed as a topological group with elements we denote ρ. Since sticky-spiders can be represented as unions of effects followed by states, we can define a binary relation on sticky-spiders that tells us whether they are the same up to rigidly displacing component shapes:

Definition 1.5.1 (Displacement relation)**.** Two sticky-spiders (cyan and green, both assumed to be nice here), each with components indexed by I, are *equivalent up to displacement* when there exist ρ_i such that:

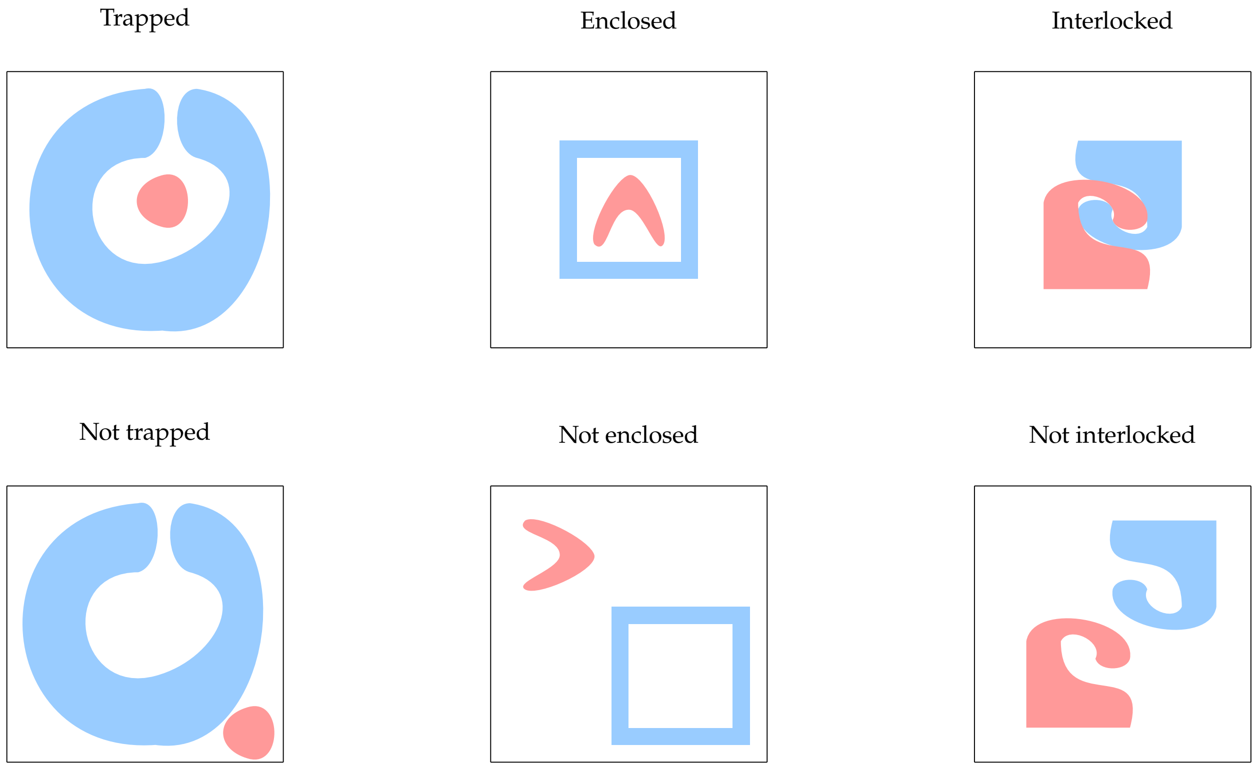


We've suppressed labelling of the states and we've contracted the cup to just depict the open state as a semi-circle.

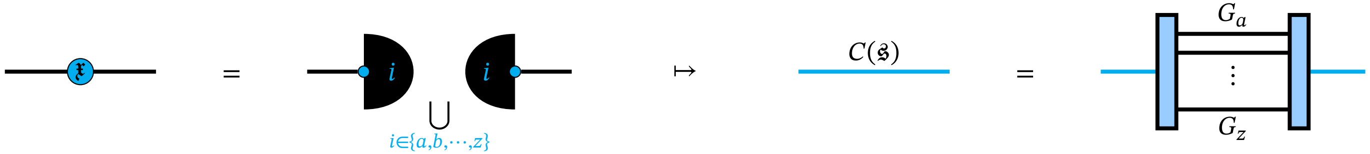
Displacement is evidently an equivalence relation, and moreover requires that the two spiders related have the same number of components. Now given a particular nice spider, we treat its equivalence class of spiders as a configuration space in which we have access to all of its rigidly displaced variants at once.

Definition 1.5.2. The *configuration space* $C(\mathfrak{s})$ of a nice spider $\mathfrak{s}$ with indexing set I is the topological space with underlying set defined to be the equivalence class $[\mathfrak{s}]$ of $\mathfrak{s}$ under displacement. Assuming the topological group of rigid displacements is itself a topological space G, the topology of $C(\mathfrak{s})$ is a restriction of $\bigtimes^{|I|} G$ to those $|I|$-tuples of displacements witnessed by $[\mathfrak{s}]$.

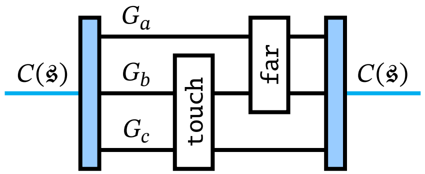
Example 1.5.3 (The connected components of configuration space)**.** Configuration space allows us to define a "slideability" relation between configurations of a spider $\mathfrak{k}$ as the endpoints of continuous functions from the unit interval into $C(\mathfrak{s})$. This in turn allows us to consider what the connected components of configuration space are. Evidently, there are pairs of spiders that are both valid displacements, but not mutually reachable by sliding. For example, shapes might *enclose* or *trap* other shapes, or shapes might be *interlocked*. So at first blush, the connected components of configuration space tells us something about holes, or the cohomology of configurations. Depicted are some pairs of configurations corresponding to some linguistically topological terms that are mutually unreachable by rigid transformations, and so must live in disconnected components of configuration space.

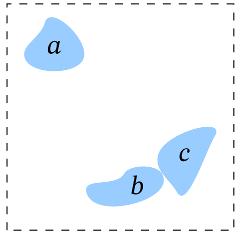


In configuration spaces we're making use of the fact that any displacement relationship comes with (up to a non-unique choice of basepoints for each component shape) a witnessing tuple of ρ_is. As a consequence, the configuration space of a sticky-spider is a retract of the product space $\bigtimes^{|I|} G$ where G is the topological group of displacements, and we can use the identity relation between the section and retraction to strip the configuration space wire, revealing each of the $\bigtimes^{|I|} G$ like guitar strings: each element of the set that the initial nice spider $\mathfrak{s}$ splits through gets its own string.

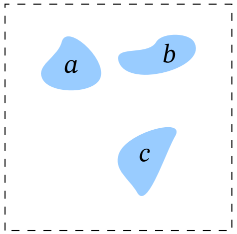


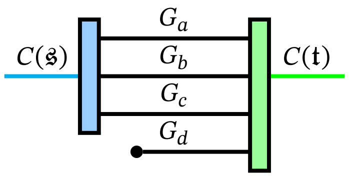
Note that although every guitar string is G, there is extra typing data indicating which element of the indexing set of the spider each G corresponds to. So here's a model in which the named wires of text circuits make sense. We can put gates on the guitar strings, which may for example correspond to constraints on the relative positions of shapes in configuration space.

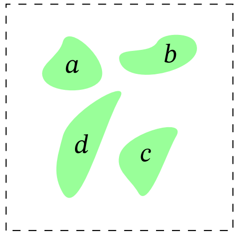




The next thing we can try is to add and subtract shapes from configuration spaces, and while there are technical details like matching choices of basepoints I'll gloss over, the gist is this: when the shapes in a nice spider $\mathfrak{s}$ are a subset of the shapes in a nice spider $\mathfrak{t}$, we can add in states to the guitar-picture of $\mathfrak{s}$ and wrap them up again using the idempotent of $\mathfrak{t}$, and we can delete wires in the guitar-picture of $\mathfrak{t}$ and wrap that up using the idempotent of $\mathfrak{s}$.







The last stop in this sketch is disintegrating and integrating shapes; if we could freely break apart a shape, we know that in principle we get another configuration space where we can manipulate those parts, and if we can glue those pieces back together again, then we could do simple things like open and close containers. Let's first define the disintegration relation between spiders. Observe that the data of a nice spider is equivalently viewed as a function $f : I \to \mathfrak{O}$, where I is the indexing set, and $\mathfrak{O}$ is some set of opens with whatever well-behaviour condition, along with the constraint that $f(x) \cap f(y) \neq \varnothing \Rightarrow x = y$ that enforces non-overlapping shapes. This perspective gives us a foothold to define a disintegration relation: a "more refined" spider is one that has a superset of I as domain, with a function that sends elements of the indexing set to either the same shape as f, or a subshape.

Definition 1.5.4 (Disintegration)**.** Let $\mathfrak{s}$ and $\mathfrak{t}$ be nice spiders, described by functions $s : I \to \mathfrak{O}$ and $t : J \to \mathfrak{O}$ respectively. $\mathfrak{t}$ *disintegrates* $\mathfrak{s}$ ($\mathfrak{t} \succ \mathfrak{s}$) if there exists a surjective $d : J \twoheadrightarrow I$ such that $g = f \circ d$, and such that for all $i \in I$ and all $j \in d^{-1}(i)$, $g(j) \subseteq f(i)$.

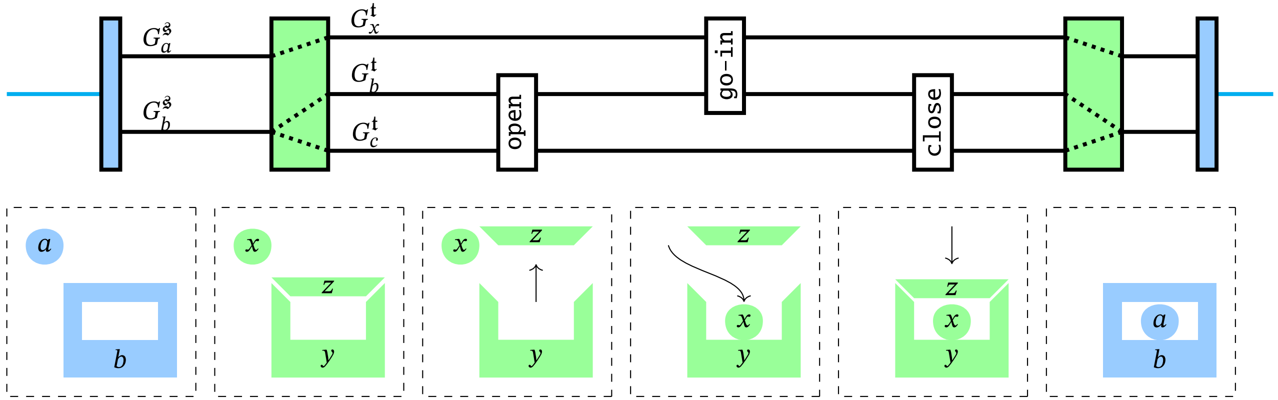
Since the composition of surjectives is also surjective and the subsethood condition is transitive, disintegration is a transitive relation. It's also reflexive, and since surjections $A \twoheadrightarrow B$ and $B \twoheadrightarrow A$ implies a bijection $A \simeq B$ and $X \subseteq Y$ with $Y \subseteq X$ implies $X = Y$, we also have antisymmetry, and hence a partial order. Treating the identity disintegration as globally minimal, we can define shatterings as locally minimal elements.

Definition 1.5.5 (*Solve*)**.** $\mathfrak{t}$ *shatters* $\mathfrak{s}$ if $\mathfrak{t} \succ \mathfrak{s}$, and for all spiders $\mathfrak{q}$, $\mathfrak{t} \succ \mathfrak{q} \succ \mathfrak{s} \Rightarrow \mathfrak{q} = \mathfrak{t}$ or $\mathfrak{q} = \mathfrak{s}$, up to bijective relabellings of indexing sets.

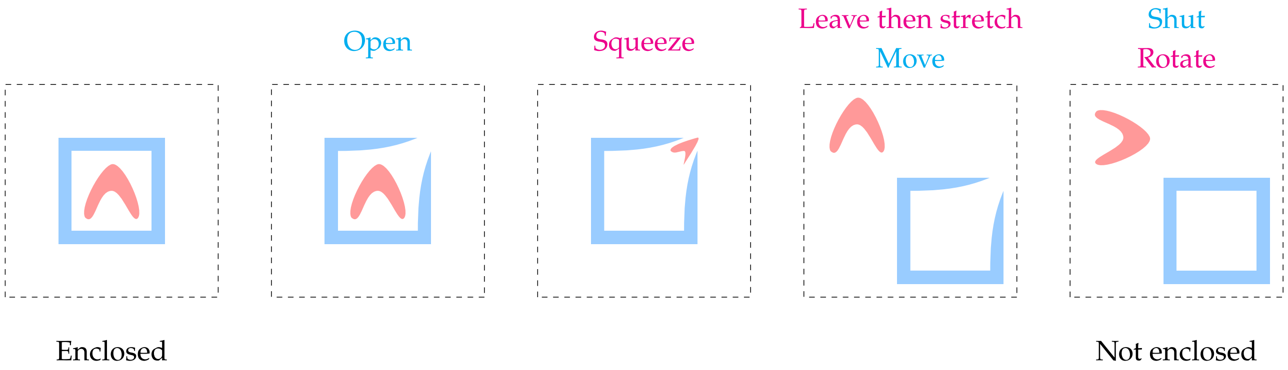
The intuition behind shattering is that the $\subseteq$-condition in the disintegration relation lets the disintegrating spider "shave a little" off of the disintegrated spider, and locally minimal disintegrations "shave the least off", doing the best they can to partition shapes. So now we get gluing for free:

Definition 1.5.6 (*et Coagula*)**.** $\mathfrak{t}$ is a *gluing* of $\mathfrak{s}$ if $\mathfrak{s}$ shatters $\mathfrak{t}$.

Example 1.5.7 (Putting something in a container)**.** To put a blob inside a container, we first shatter the container of the initial spider $\mathfrak{s}$ to obtain a new spider $\mathfrak{t}$ that expresses the container as a combination of a container and a lid, then (implicitly using dynamic verb composition of terminatives) we can move the lid, put the blob in, close the lid, and glue. Below the circuit we represent one possible series of consistent snapshots as a vignette, out of the many possible series of configurations that satisfy our linguistic description above.



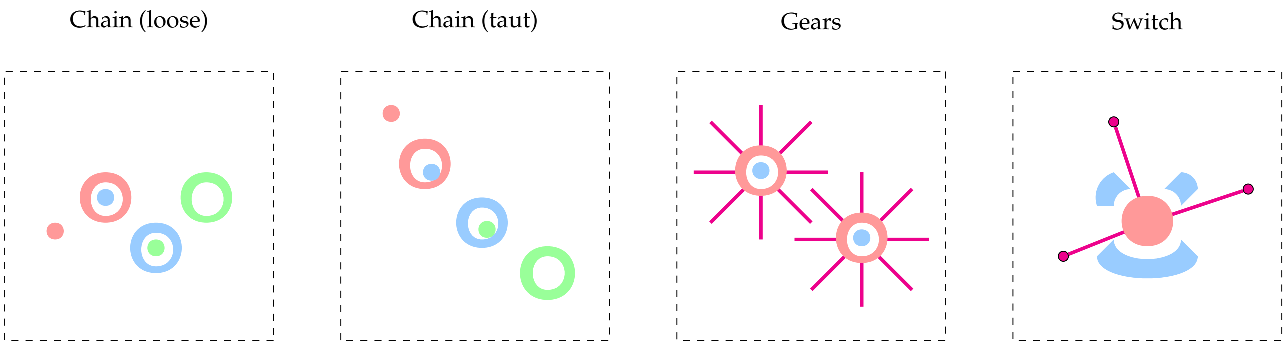
In principle, shapes can be shattered arbitrarily finely, which permits us some degree of freedom in specifying how a container opens. In conjunction with a topological group of transformations that includes scaling, we may express different ways in which things get in and out of containers, or otherwise leave the original connected component of configuration space they start in. Here again I'm colour coding different shapes of the same spider with different colours.



I'll close this sketch with something cute: if manipulating shapes in configuration space is serious and sensible stuff, then just about anything is. We can (ab)use the fact that shapes of a nice sticky spider do not overlap to model mechanical components, where acceptable configurations of different shapes are mutually constrained in a productive way. In particular, this means we may consider any linguistic semantics grounded in mechanical or boardgame-tabletop models to be formal: in principle anything that can be represented by mechanisms and meeples is fair game. This gives us some cool possibilities for formal models of natural language, as there are a lot of mechanical models, including: clocks [duh.], analogues of electric circuits [noa], computers [Ric15], and human-like automata [wik22].

Objection: Isn't this way outside the scope of formal semantics? Insofar as semantics is sensemaking, we certainly are capable of making sense of things in terms of mechanical models and games by means of metaphor, the mathematical treatment of which is concern of Section 1.6. It's probably the case that any definition that encompasses what's going on here as formal semantics would also have to consider the programming of a videogame to also be a form of formal semantics; personally I think that's ok, because I don't consider any particular form of mathematics-as-methodology to be privileged over others. Feel free to disagree.

Example 1.5.8 (Mechanical semantics)**.** Here I'm going to allow shapes to be unions of disjoint contractibles, and I'll colour-code the different shapes in the spiders differently so the different components are clear:



1.6 Formal models from figurative language

Figurative language is when language is used non-literally, e.g. `to bathe in another's affection`. Figurative language subsumes analogy (`built like a mountain`), metaphor (`she got a lot out of that lecture`) and some idioms (`raining cats and dogs`). The issue with figurative language for formal semantics, insofar as formal semantics is concerned with truth-conditions, is that one requires an underlying model in order to begin truth-conditional analysis. The role of figurative language, especially that of metaphor, is in some sense to provide those models in the first place, so the truth-theoretic (or inquisitive, or dynamic) approach to semantics operates at an inappropriate stage of abstraction. We might illustrate or depict schema to represent figurative language, but to the best of my knowledge, there is no formal account of how the systematicity of a chosen schematic corresponds to the organisation of a metaphor or concept. So what is required is a methodology to construct the underlying models from the figurative language in a more-or-less systematic way.

[Red] argues convincingly that the metaphor `IDEAS are CONTAINERS` is pervasive in English; it is just about the only way we talk about communication. Yet there is no literal sense in which one can 'get something' out of a lecture or 'pack a lot' into a book. Evidently the *systematicity of the metaphor itself* yields the common structure from which we can even begin to consider pedestrian truth-conditional analyses; i.e. language has a role to play in constructing the stage, and afterwards we can reason logically about the actors and events. The process by which language constructs the underlying model is not by nature truth-conditional: there is no fact of the matter before it is read of a fictional character's eye-colour, but it does become a facet of the reader's world-model afterwards. Therefore: of which truth-conditions cannot speak, thereof truth-conditionalists must remain silent.

The whole point of mucking around with **ContRel** earlier is this: figurative language can be formally interpreted as vignettes involving topological figures. I will demonstrate here that cofunctors from **ContRel** into text-circuits representing utterances are promising candidates for the formalisation of figurative language. My focus will exclude idiomatic language and one-off analogies in favour of metaphor just because the latter is most interesting, though the methodology applies in other cases of figurative language. I will take a *metaphor* to be figurative language that utilises the systematic structure in one conceptual domain to give partial structure in another conceptual domain. This may subsume some cases of what would otherwise be called *similes* or *analogies*. The differences far as I can tell between a metaphor and an analogy is the presence of systematicity in the former, and a weak requirement that the correspondence involves separate conceptual domains. It doesn't really matter for this discussion what the difference is.

Meta, beyond. Phor, as in amphora, an agent, carrier, or producer. Metaphor carries meaning beyond one domain to another. It bears and produces meaning. Metaphors are the primary agents of meaning.

First, we observe that we can model certain kinds of analogies between conceptual spaces by considering structure-preserving maps between them. For example, Planck's law gives a partial continuous function from part of the positive reals measuring tempature of a black body in Kelvin to wavelengths of light emitted, and the restriction of this mapping to the visible spectrum gives the so-called "colour temperature" framework used by colourists. It will turn out that a decategorified cofunctor has the right kind of structure.

Second, we observe that we can use simple natural language to describe conceptual spaces, instead of geometric or topological models. Back to the example of colour temperature, instead of precise values in Kelvin, we may instead speak of landmark regions that represent both temperature and colour such as `incandescent` and `daylight`, which obey both temperature-relations (e.g. `incandescent is cooler than daylight` and colour-relations (e.g. `daylight is bluer than incandescent`).

Third, we observe that we can also use simple natural language to describe more complex conceptual schemes with interacting agents, roles, objects, and abilities. This will require a cofunctor. Organising this linguistic data in the concrete structure of a text circuit allows us to formally specify what it means for one conceptual scheme to structure another by describing structure-preserving maps between the text circuits. This will allow us construct topological models of metaphors such as `TIME is MONEY`.

1.6.1 *Temperature and colour: the Planckian Locus*

Example 1.6.1 (The Physicists' Planckian Locus)**.** Planck's law describes the spectral radiation intensity of an idealised incandescent black body as a function of light frequency and temperature. Integrating over light frequencies in the visible spectrum yields a function from temperature of the black body to chromaticity.

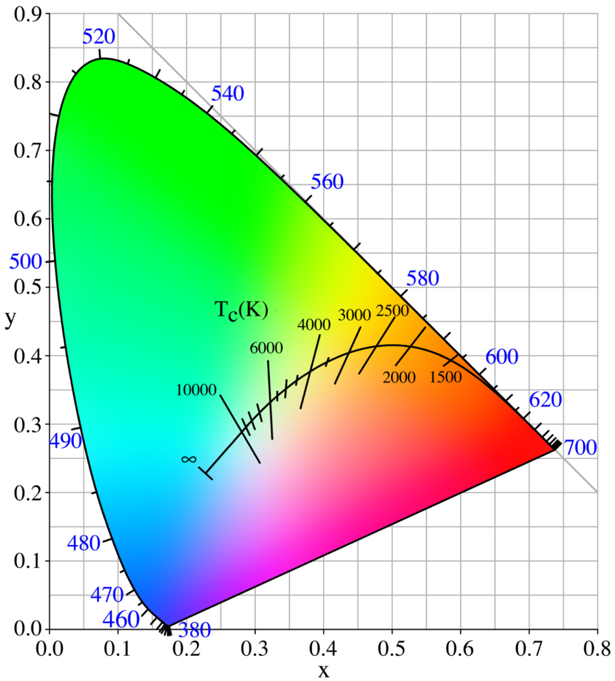


Figure 1.11: The Planckian Locus in the CIE 1931 chromaticity diagram. Chromaticity refers only to the hue of a colour, without other domains such as saturation.

Abstractly, the Planckian Locus is a continuous function mapping the positive real line representing the conceptual domain of temperature into the plane representing the conceptual domain of colour. The Planckian locus is the basis of colourist-talk about colour schemes in terms of temperature, which allows them to coordinate movements in colourspace using the terminology of temperaturespace, e.g. `make this shot warmer`. This fits with what we would prototypically expect a metaphor to allow us to do with meanings.

However, the particular mathematical conception of metaphor-as-map in Example 1.6.1 is too rigid: it only goes one way. It is a specific and inflexible kind of metaphor that does not behave at all outside its specified boundaries. For example, colourists have to deal with offsets towards green and magenta, which are not in the chromaticity codomain of the function given by Planck's law. It would be truer to life if we further analysed the function as mediated by a strip.

Example 1.6.2 (The colourist's Planckian Locus)**.** Now we aim to extend our mathematical model to accommodate the fact that colourists deal with chromatic offsets or deviations from the mathematically precise locus given by Planck's law.

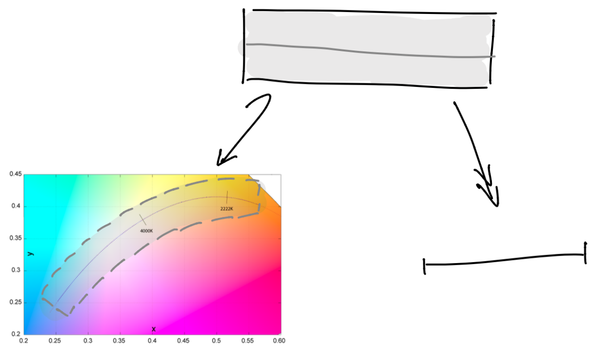


Figure 1.12: Consider the unit square (depicted as a strip) as a fiber bundle over the unit interval representing temperature range. There is an injective continuous map from the strip into colourspace that is centered on the Planck Locus.

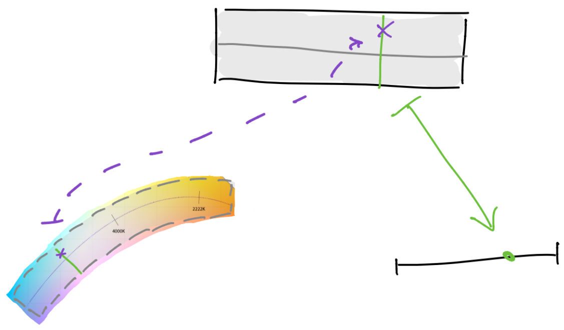


Figure 1.13: The left leg is bijective in the image restriction, so any point or displacement in the offset-strip in colourspace can be lifted to a point in the apex strip, which is then projected down along with other points in the vertical fiber to a point in temperaturespace. So we have a decategorified cofunctor!

A refinement we have just captured is the partially-structuring nature of metaphor [LJ03]. In the language of our running example, pure green is outside the scope of the colour-temperature correspondence given by the Planckian Locus, so the metaphor is only a partial structuring of the colour domain according to the temperature domain. This partiality in the colour domain means that it would have been inappropriate to model the passage of colour-talk to temperature-talk as a function from colour to temperature, as functions are total, rather than partial, on their domain. While it is conceptually nice that we are on the way to recovering monoidal cofunctors as a model of metaphor, why didn't we stay simple and just use a partial function? The answer is that the strip at the apex represents the *talk* part of colour- and temperature-talk.

Example 1.6.3 (Conceptual transfer between domains)**.** When colourists use the temperature metaphor they might say "hot", "warm", "cooler", which are not specific temperature ranges in Kelvin, but concepts in temperature-space. Recalling that we may consider concepts to be open sets of a topology (and comparatives as opens of the product), we observe that we can linguistically model regions on the positive reals with words `little` (labelled l), `lot` (labelled L), and `more` (labelled M), an algebraic basis from which derive `less` by symmetry, and other regions such as `more than a little, less than a lot`. In this particular running example, it happens that both legs of the span of functors have a lifting property, which explains how we might model the fact that conceptual colourist-talk of "daylight" or "candlelight" in the colour domain can be sensibly interpreted in the temperature domain. The formalisation of this fact follows by symmetry from this example.

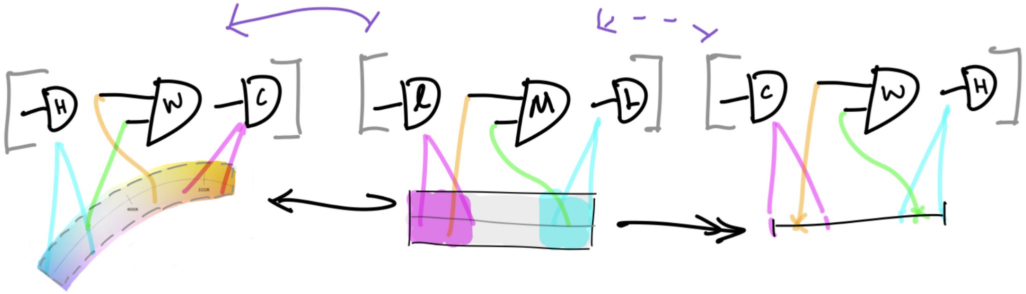


Figure 1.14: Starting from the right, the lifting property of the right leg is what lets us map "hotter and colder" temperature talk into the more abstract quantity-talk of "more and less" in the apex strip. Then the left functor sends quantity-talk into the colour domain, which allows "hotter and colder" to be used in the colour domain.

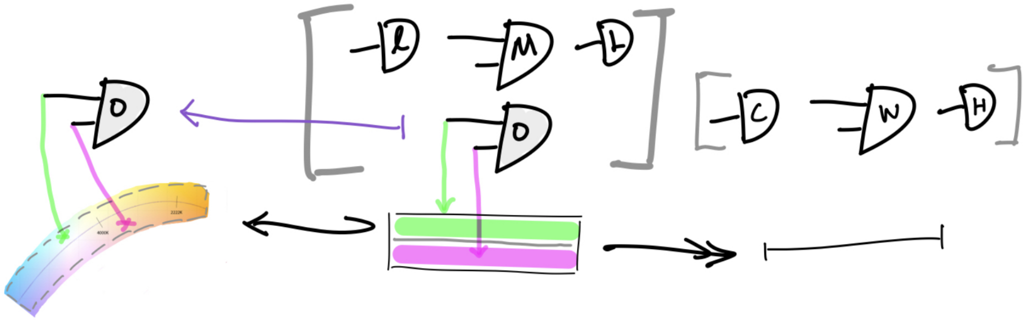


Figure 1.15: The additional expressive power that the apex strip gives is the concept of vertical offset, which doesn't appear in the real line. So the apex strip allows talk of quantity and offset, and this offset, when translated into the colour domain, allows talk of offset towards and away from, for instance, green.

1.6.2 Time and Money: complex conceptual structure

Metaphor is perhaps the only methodology we have for making sense of certain abstract concepts, such as Time. For example, many languages make use of the metaphor `TIME is SPACE`, in which space-talk is used to structure time. In English, the future is ahead of us and the past behind, while conversely, for the Aymara the future is behind and the past is ahead. Orthogonally, in Mandarin the future is below and the past is above. We have already demonstrated that we have the tools to deal with conceptual transfer between static conceptual spaces viewed as topological spaces via spans of continuous maps. What is of concern to us are *dynamic* metaphors that involve a conceptual space-*time* with agents and capabilities and so on. The following discussion draws heavily from [LJ03].

[2] [

For example, in English, we make ample use[2]]To our detriment. We could just as well have chosen the metaphor `TIME is FOOD`, which provides a liberating sense of mastery (at least in a context where food is abundant): time can be prepared, produced, consumed, spiced if dull and best shared with loved ones. of the metaphor `TIME is MONEY`. There two mathematically relevant aspects of metaphor that I want to draw attention to for this metaphor. Firstly, that the conceptual affordances money-talk is marshalled to give structure to time-talk, where there is no such structure were it not for the metaphor. Secondly, that metaphor has a partial nature, in that it is not the case that the metaphor licenses all kinds of money-talk to structure time-talk.

To establish the first point of conceptual transfer, a phrase like `Do you have time to look at this?` is completely sensible to us, but literally meaningless; even if we had an oracle to measure possession, what would we point it at to measure a person's possession of time? Even if we accept some argument that the concept of possession is innate to the human faculty, when we say `This is definitely worth your time!` or `What a waste of time.`, we are drawing upon value-talk that is properly contingent in the socially constructed sense upon the conceptual complex of money.

[3] [

To establish the second point of partiality, consider that money can be stored in a bank, whereas there is no real corresponding thing in the common conceptual vocabulary which one can store time and withdraw it for later use[3]]Although, in a wonderful example of 'pataphysical thinking, "Time Banks" have existed since the 19th century, which are practices of reciprocal service exchange that use units of time as currency.. But the partiality constraint is itself partial. For instance, one can invest money into an enterprise in the expectation of greater returns, and this is not appropriate for many domains of time-talk, but there is a metaphorical match in some specific contexts, such as text-editor-talk: `learning vim slows you down at first but it will save you time later`.

Now I'll try to demonstrate by example that the kinda-cofunctors we explored in Section **??** between text circuits do all of the things we have asked for. The components of text circuits serve as an algebraic basis for dynamic conceptual complexes, while the kinda-cofunctor handles partial structuring of one conceptual domain in terms of another.

Example 1.6.4 (`Vincent spends his morning writing`)**.** To begin a formal figurative interpretation via the metaphor `TIME is MONEY`, we require some model of the conceptual domain of money, as well as a topological interpretation. As a first pass, we understand that money can be exchanged for goods and services, so we will settle for a text-circuit signature for trade to serve as the conceptual domain as the apex of a cofunctor, given in Figure 1.16. The elements of the topological model are given in Figure 1.17. The behaviour of the opfibration part of the cofunctor is detailed in Figure 1.18, and that of the identity-on-objects functor in Figures 1.19, 1.20, and 1.21. The figurative model serves as a foundation from which truth-theoretical semantics can begin. In the sketched interpretation, there aren't too many interesting questions one can ask, but the purpose of this example is to point out that in principle, we can exploit the systematicity of metaphor by constructing figurative mechanical models for which interesting questions can be asked and answered truth-theoretically, as in Figure 1.22.

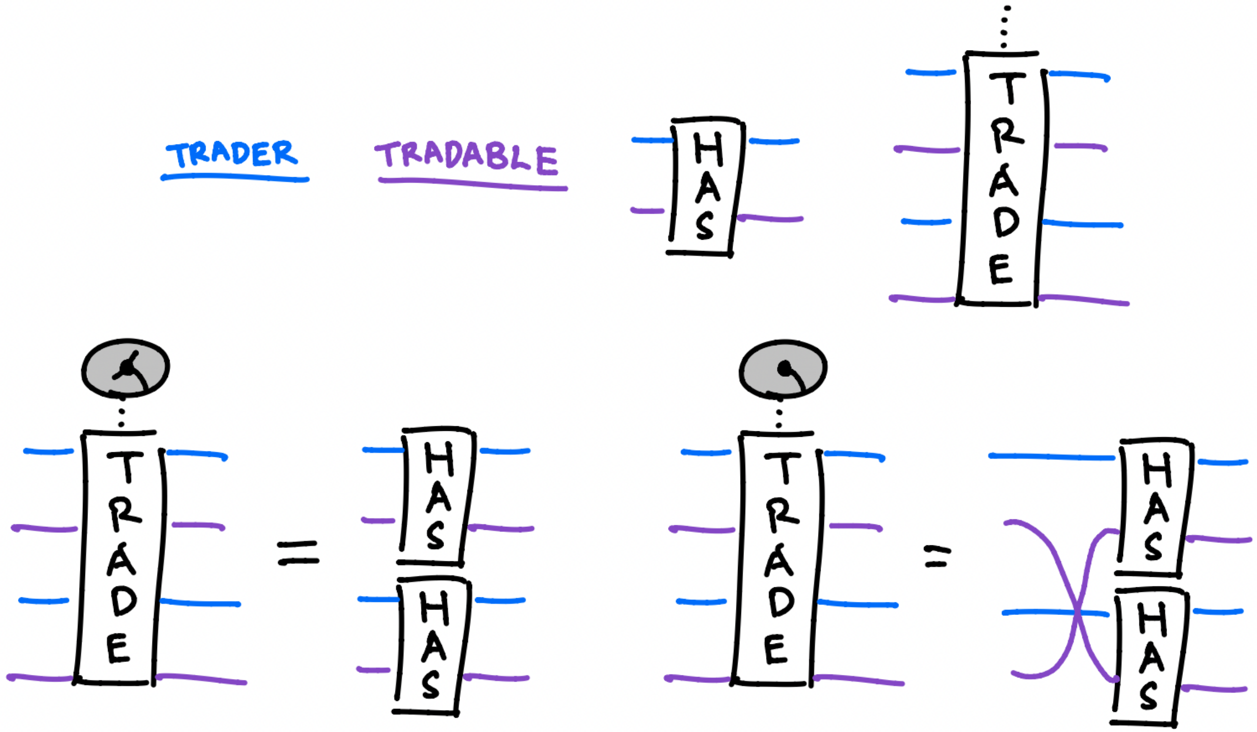


Figure 1.16: In the `TRADE` signature, we define two roles as wires: `TRADERS` and `TRADEABLES`. There is one static relation `HAS` to indicate a trader's ownership of a tradeable, which can be further elaborated with equations to indicate e.g. exclusivity of ownership by interpreting violations of exclusivity as a zero morphism, assumed but elided for brevity. There is one dynamic verb (treated as a homotopy) `TRADE`, which at time 0 enforces a precondition that the traders have their respective tradables, and at time 1 (completion of the trade), the traders swap possession of their tradeables. The `TRADE` signature contains all nominal instantiations of nouns with respect to roles, which will be illustrated shortly.

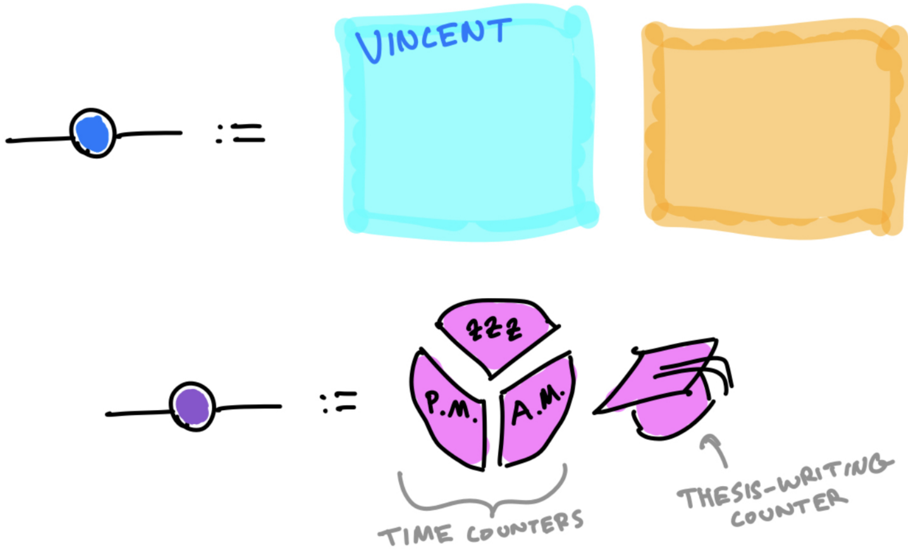


Figure 1.17: We build the topological model from two sticky spiders in the Euclidean plane. The TRADER spider will distinguish two regions of possession, so that HAS may be interpreted as a region-test. The TRADEABLES spider will specify four meeples or counters, three for time, and one for thesiswriting; we will use the configuration space of the TRADEABLES spider to regulate their movement and distribution.

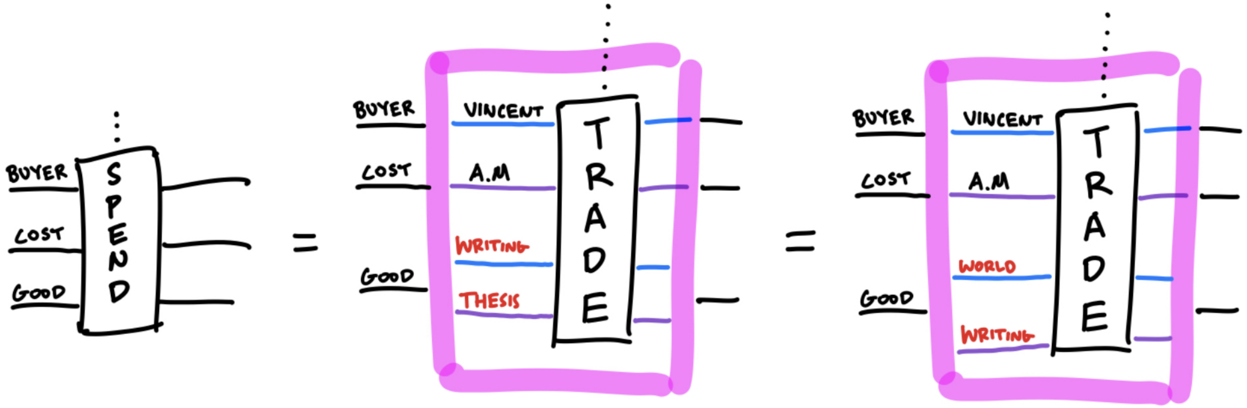


Figure 1.18: Next, we have to specify what the discrete opfibration is doing. Recalling our functor box notation, we can consider the job of the discrete opfibration to be role assignment from the verb SPEND in the utterance to the verb TRADE in the conceptual domain. The opfibration forgets about role-assignments in its domain by sending them to the monoidal unit. The lift of the opfibration is a role-assignment. (Arguably) unambiguously in this example, `Vincent` is the spender and the first trader, and `A.M` is the cost and the first tradeable. However, there are two options to resolve `writing` treated as a noun-phrase in the role of GOOD. In the first lift, `writing` is resolved as the other trader, and the implicit good as `thesis`. In the second lift, `writing` is the tradeable and something else is the

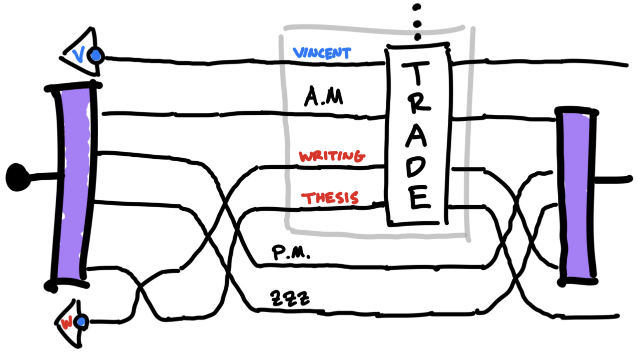


Figure 1.19: The section of the opfibration over the SPEND verb is a tabulation of all the ways in which conceptual roles in the TRADING domain can be assigned. To continue the example, we will assume the first lift in Example 1.18 as our interpretation. The identity-on-objects functor part of the cofunctor maps our chosen interpretation into the following diagram in **ContRel**. The configuration space of the TRADEABLES spider is expanded via split idempotent so that all thin wires in the diagram are typed as the Euclidean plane. Recalling Example 1.1.1, HAS is interpreted as the intersection of the position of a counter with the possessive region of the respective trader.

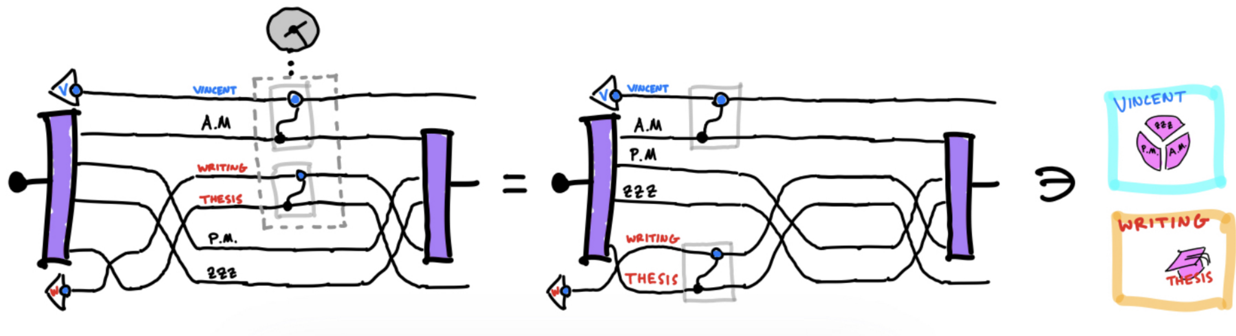


Figure 1.20: We may verify that the equations governing TRADE cohere with our topological figures. At time 0, before the trade, we can calculate that the permissible figures have Vincent in possession of A.M and writing in possession of thesis.

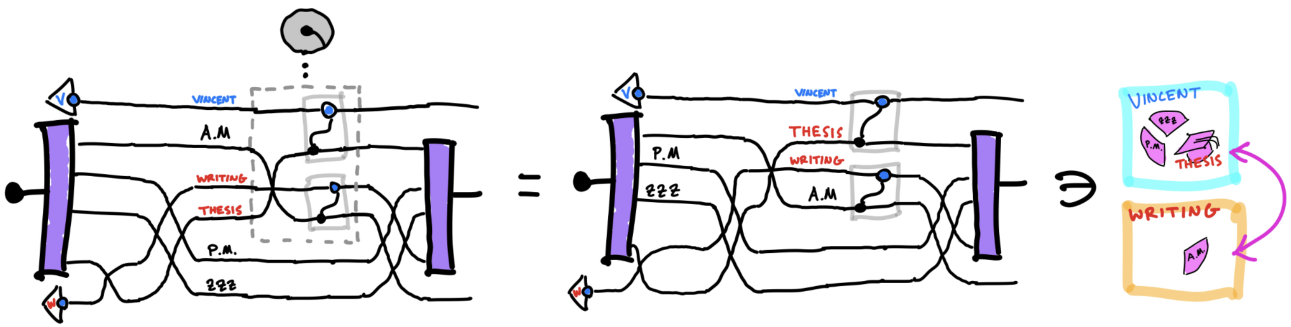


Figure 1.21: At time 1, we may calculate that the permissible figures must be such that `Vincent` is in possession of `thesis` and `writing` is in possession of what was previously my morning.

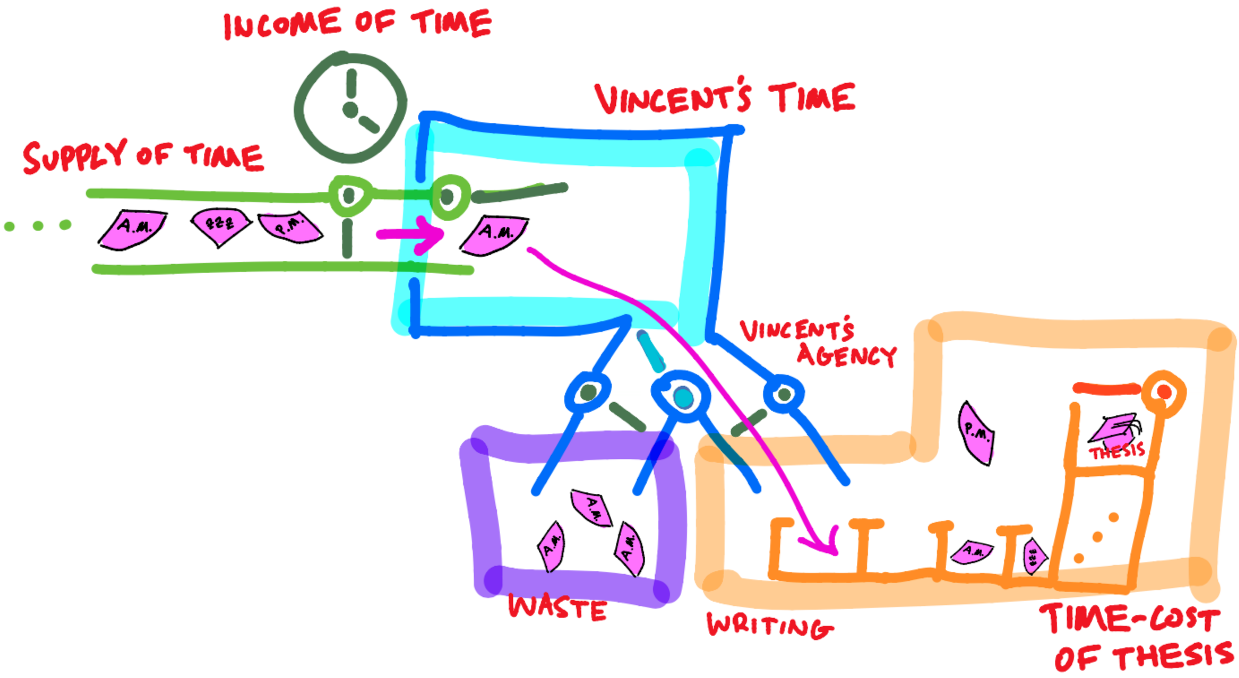


Figure 1.22: In a more detailed conceptual model of `TIME is MONEY`, rather than just `TRADE`, we might consider income, the spender's agency, and cost. In Euclidean 3-space, we might model income as a clock-gated mechanism that deposits time-tokens serially into Vincent's possession, along with his agency as a gated chute, and the time cost of writing a thesis as a dispenser that requires a certain number of tokens to release a thesis-token into Vincent's possession. In this sketch model, one obtains short films for `He used to waste his mornings but now he spends them writing`, or `He once spent an evening writing but made no progress`. One can then ascertain certain consequences truth-theoretically; for instance that there is at least one morning that was not spent on writing, or that there is at least one evening spent on writing but not inside the slot that would help a thesis-release mechanism trigger. In every case, cofunctoriality handles bookkeeping for role-interpretation choices and guarantees systematicity of the topological figure according to the signature at the apex model that models the organising

Objection: Hold on, that last figure wasn't justified. It's justified in principle.

Objection: On what principles? Good question!

1.7 The logical conclusions

Like how the Church-Turing thesis is a declaration of the limits of computability by informed consensus, we might ask what idealised theses we might posit. Now we recap what we've seen.

Review of Chapter 1. We demonstrated that by encoding meaning-relations as topological connectivity (which is in a broad sense the whole point of string diagrams as a bridge between algebra and topology), we can capture systematic structural relationships as configurations of functors. In particular, we were able to diagrammatically reason about the correspondence between a pair of simple productive and parsing grammars subject to the evident constraints of communication. With some imagination, there are two theses here to obtain, or declare:

Thesis 1.7.1 (String diagrams for composition)**.** Compositionality equals topological representability; in particular, meaning relations in text are witnessed by connectivity of string diagrams.

Thesis 1.7.2 (String diagrams for systematic relationships)**.** Systematicity equals functorially witnessed relations; in particular, spans of functors between families of string diagrams witness agreement between different theories as topological equivalence.

Review of Chapter 2. We demonstrated that from the mathematical perspective of weak n-categories, there is no fundamental difference between a broad class of productive grammars at a structural level, including all string-rewrite systems, tree-adjoining grammars, and transformational grammars. In this setting, we created a circuit-growing grammar and demonstrated a correspondence – the Text Circuit Theorem – between text circuits and grammatical text for a fragment of English, hence justifying the naïve composition of text circuits as a generative grammar. We moreover indicated how the correspondence could be expanded to cover larger fragments of English, and indicated how the particular form of text circuits made them preferable for practical application. We also mentioned that in some sense the correspondence wasn't important from the perspective that text circuits were algebraic jazz for text. Putting together the views from Chapters 0, 1 and 2, productive grammars are just ways of generating topological data, and the only practically interesting constraint about whether a productive grammar is suitable for natural language stems from whether it is paired with a parsing grammar that systematically agrees on semantics up to topological equivalence – i.e. precisely the kind of equivalence that string diagrams are invariant under. So we might reach for:

Thesis 1.7.3 (String diagrams for syntax)**.** Syntax equals a coherent method of synthesising *and* analysing composition; in particular, any internally consistent conception of natural language syntax in terms of string diagrams is permissible.

Review of Chapter 3. We give content to string diagrams by interpreting them in the category of continuous relations **ContRel**. We string-diagrammatically characterised labelled collections of disjoint shapes, along with processes that allowed us to puppeteer them in space, and test for topological relationships between shapes and places such as containment and touching. In particular, since we defined rigid motions and spatial-exclusivity of shapes, we have covered enough to linguistically specify any mechanical model up to topological invariance, and it is no difficulty to see how this setting may be expanded to include distances, directions, and forces. We saw how **ContRel** naturally gives voice to the structure of text circuits as higher-order modifications, and we also saw how **ContRel** permits us to model simple intensions via containers that mirror the space around them, which also yields a novel mathematical model of **FinRel** equipped with a Turing object, which is a mathematical model for universal computation and, in the linguistic setting, syntactic polymorphism. On the faith that any consistent and interesting system of string diagrams has some consistent and interesting computational reification, we might integrate the views of the previous chapters to obtain:

Thesis 1.7.4 (String diagrams for semantics)**.** Semantics equals computatation; in particular, any consistent computational interpretation of the content of string diagrams is permissible.

1.7.1 How things could be.

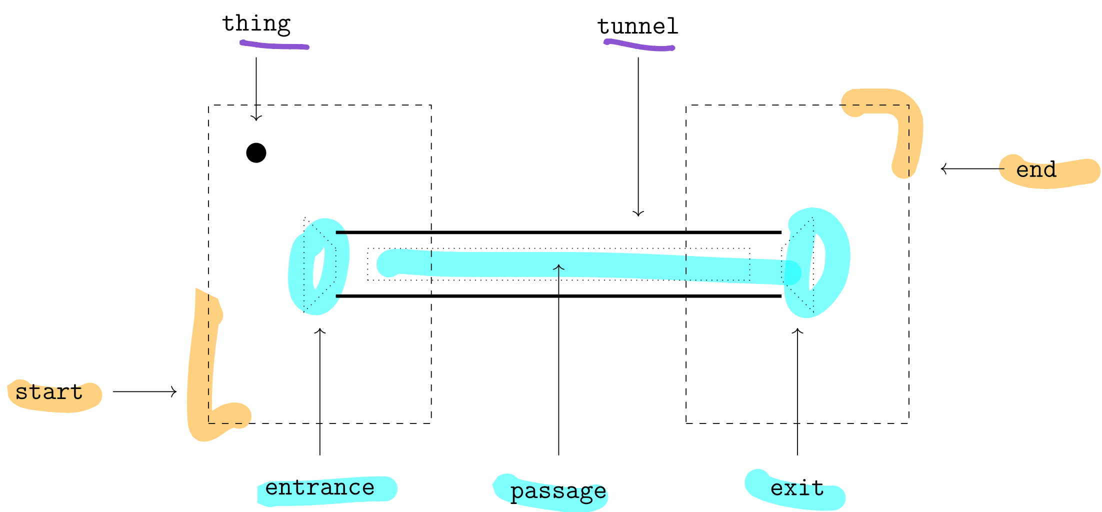
Thesis 1.7.5 (String diagrams for text)**.** String diagrams suffice for formal linguistics.

That's the best-case scenario if we accept all of the best-case scenarios above. What would such a hypothetical world look like? Recall that I still owe a worked example of computing a metaphor. Sadly, I think in a best-case hypothetical world, working out the conduit metaphor would be a kind of dull problem sheet for undergraduate mathematical linguists, who may well ask "what's the point?". Indeed, what is the point of learning all of this mathematically complicated machinery to do something everyone can do effortlessly? I'm not sure of the answer, but I do know that turning language into pictures by turning language into pictures was good fun.

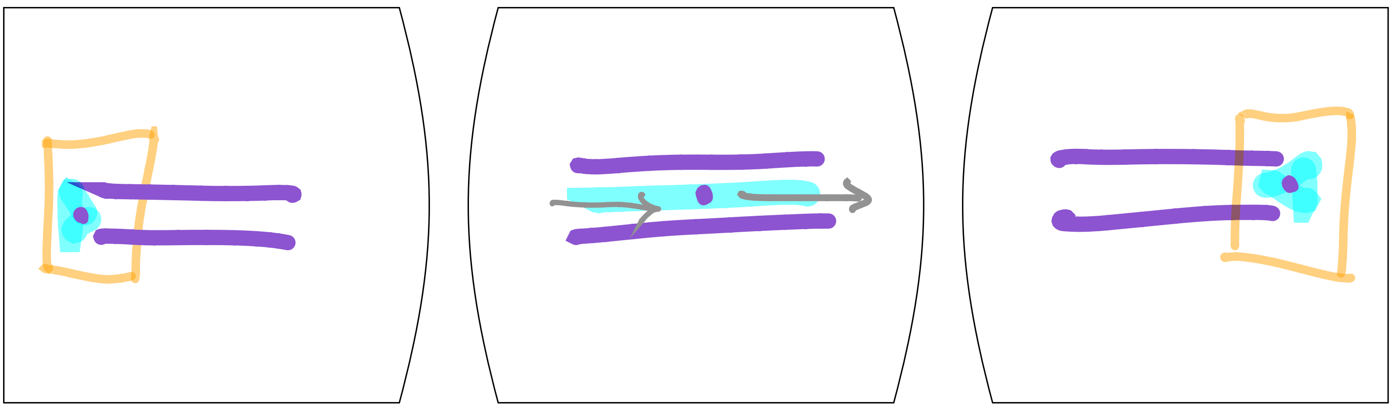
Prof. [redacted]
[redacted]: Diagrammatic Methods in [redacted]
Date: [redacted], 20[redacted]

Problem Sheet [redacted] (due Week [redacted])

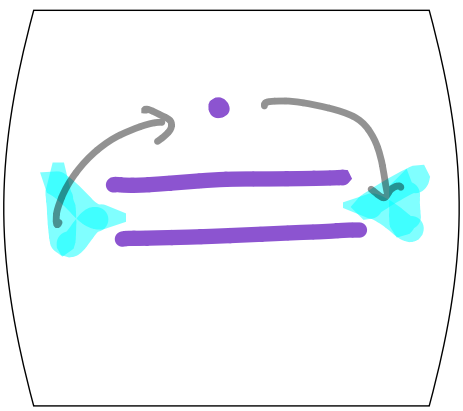
Question 1.
You may assume standard semantics of motion.
Consider the following iconic signature for the `TUNNEL` concept, with two movable shapes (in bold) and two place-indexings (indicated by dashed and dotted lines.)



a) Fill the 3-panel vignette in the `TUNNEL` signature for `thing goes through the tunnel`.



b) Depict an intermediate panel such that its splice **is not** a 3-panel vignette for the same sentence.

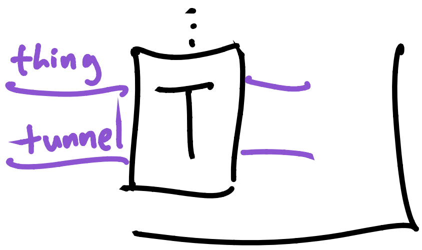


c) Briefly justify your answer for part b).

Through $\Rightarrow$ Across (start $\to$ end), but converse doesn't hold.

d) Hence, or otherwise, provide process typing and relations for `through` in `TUNNEL`.

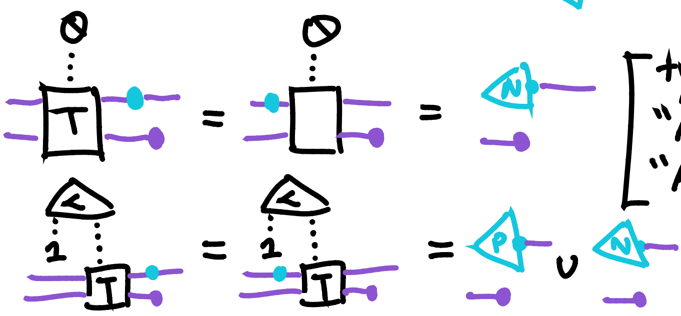
Assuming through ds dynamic in config space of {thing\tunnel}

type is: 

we want thing to move in (start & entrance) ↦ pass. ↦ (exit → End) order.

Assume ent. ⊂ start, i.e. 

[symmetrically for exit ⊂ end]

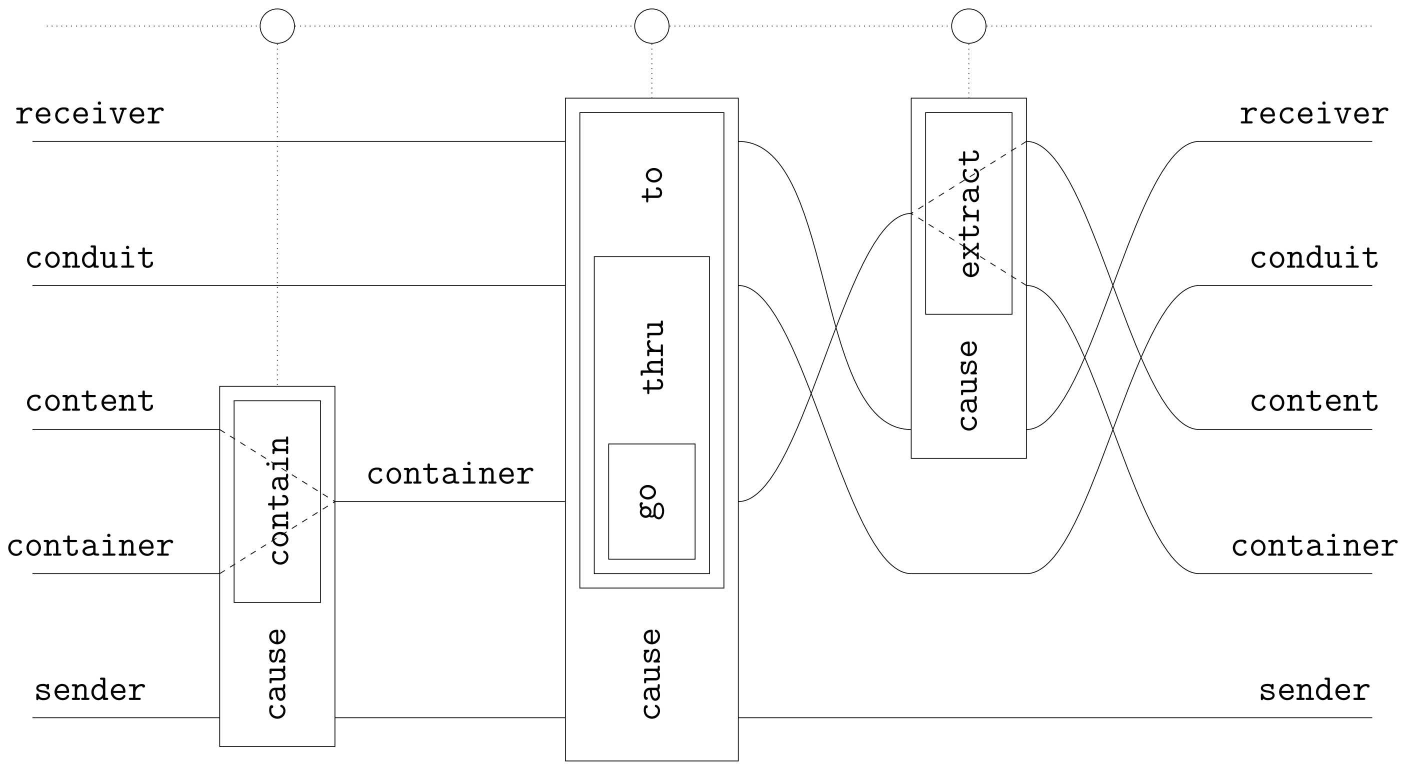


[typing by config-space of rigid mot, in plane.
"At time 0 thing is at entrance",
"At time 1 thing is at exit (by sym.)]

["At time < 1 thing is either at ent. or in passage"
This rules out countex. in part b) of Q.]

Question 2.

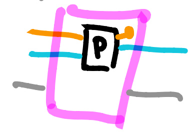
Consider the following to be the characteristic process of the `CONDUIT` metaphor.



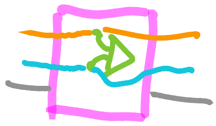
a) Analyse and equationally characterise (N)cause(contains) in terms of the dynamic verb put and static relation in. You may assume put = cause(in) and standard put-get typing. ← will use update-struct.

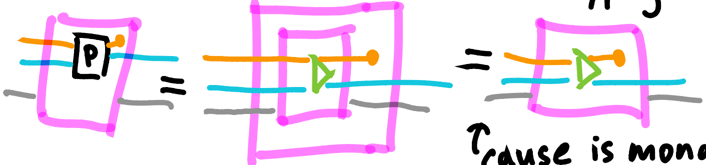
substitute

N cause container contains content = N cause( ) =



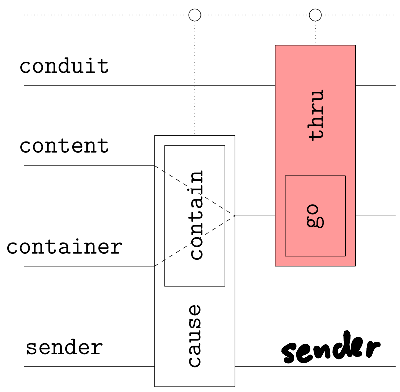
put = cause (in) → typing N put(cc) = N cause(c in c) =



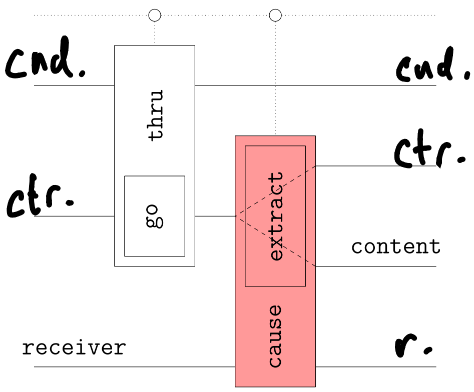


↑ cause is monadic; cause(cause(x)) = cause(x).

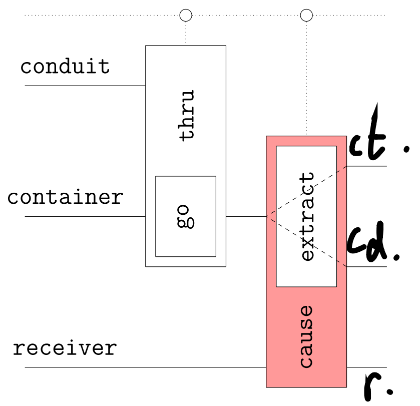
b) Assuming standard negation semantics, label and gloss the following composites.



put in but not go through.
^ container



gone through but (could/did) not extract from container.



container goes through. contents extracted but not by receiver (e.g. falls out)

c) For each composite, give an example of a matching sentence in the CONDUIT metaphor, along with with brief justification.

1. The painstaking speech fell on deaf ears.

J: painstaking speech ⇒ ideas put in words   deaf ears ⇒ words did not go through.

2. They heard him, but they didn't listen.

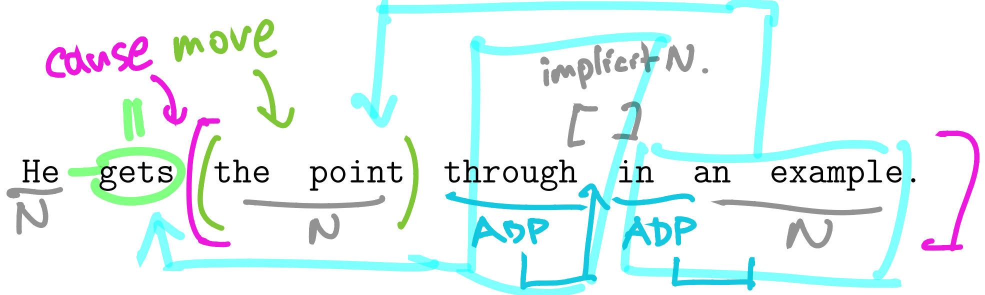
J: heard him → words came through   didn't listen → ideas/meaning not extracted from words.

3. His message was crystal-clear.

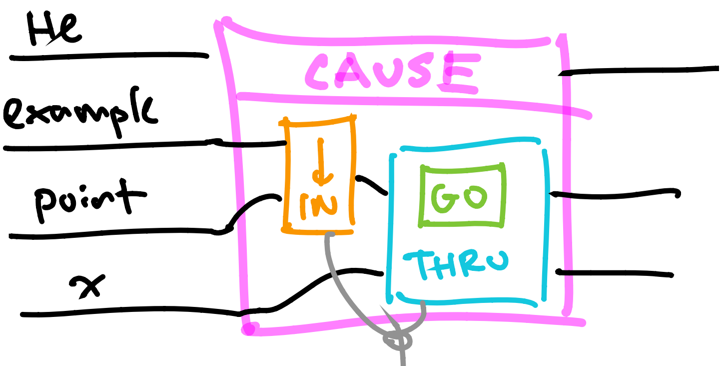
J: crystal-clear ≡ transparent → contents extracted without effort (visible)

Question 3.

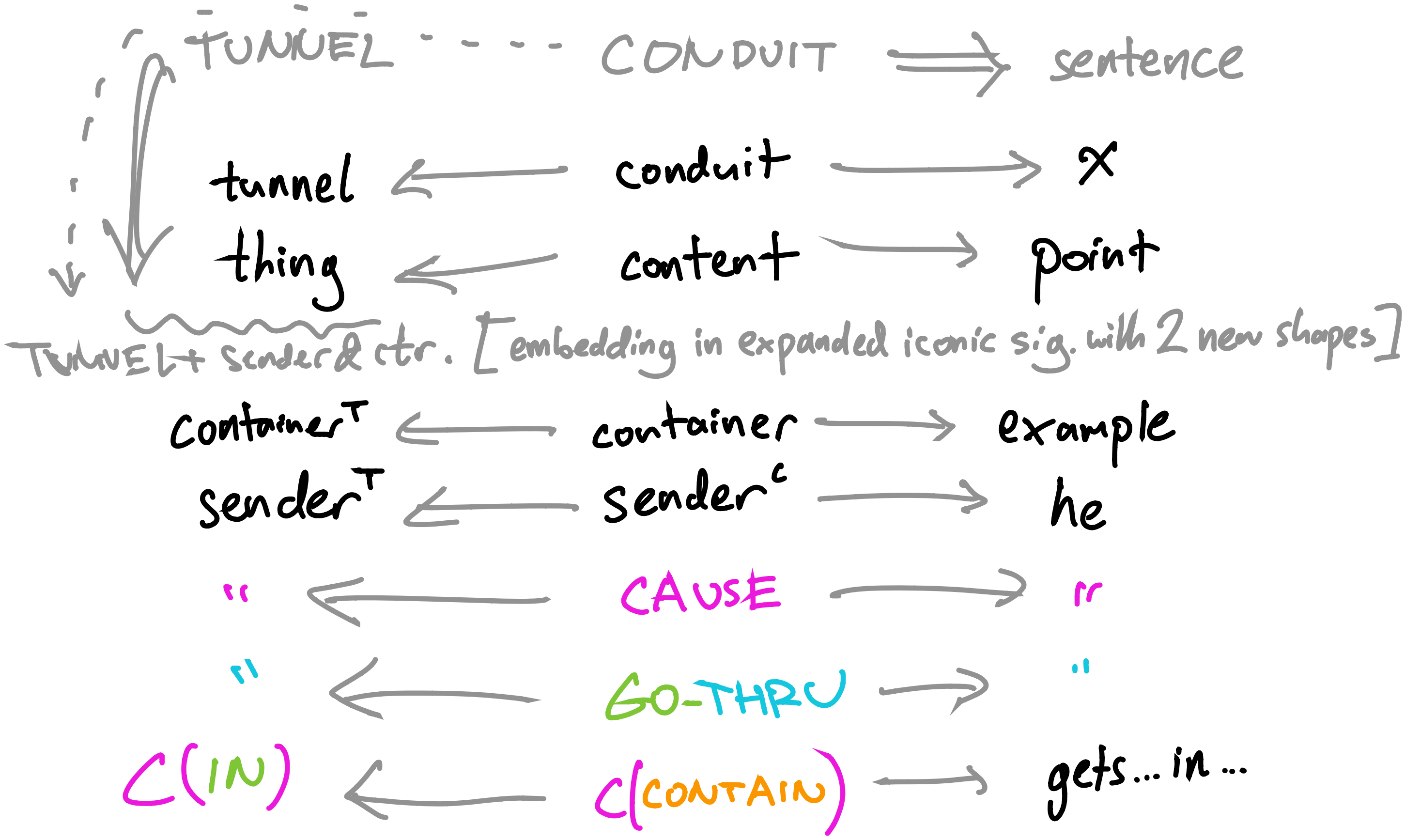
a) Provide a phrasal analysis. You may ignore contextual determiners.



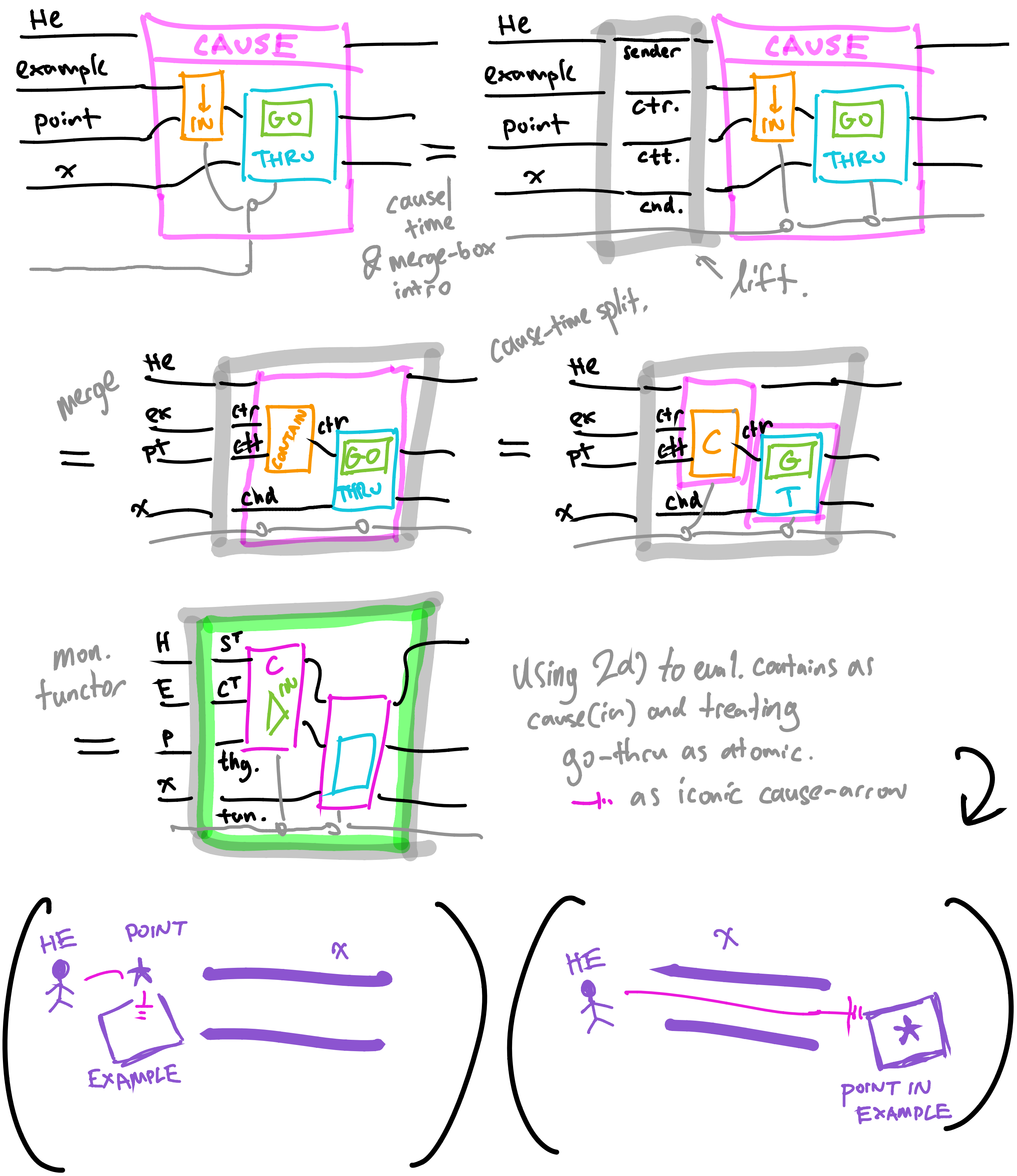
b) Hence, or otherwise, provide a text circuit for the sentence.



c) Using your answers from previous questions, compute the iconic semantics in TUNNEL by merge-boxes. You may assume standard causal semantics and notation. Justify nonstandard notations.



(An additional blank page is provided for calculation, should you need it.)



Notices: Due to an ongoing [redacted] weather-event there is a [redacted] infestation in Hall [redacted]. Next week's lectures will take place on [redacted]-[redacted]: a reminder that non-[redacted] students must have their [redacted]-modules [redacted]ed and [redacted]ing AT ALL TIMES WITHOUT EXCEPTION.

[redacted] UNIVERSITY

2
Bibliography

[CH08] J. R. B. Cockett and P. J. W. Hofstra. Introduction to Turing categories. *Annals of Pure and Applied Logic*, 156(2):183–209, December 2008.

[CK17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge, 2017.

[Coe21] Bob Coecke. Compositionality as we see it, everywhere around us, October 2021. arXiv:2110.05327 [quant-ph].

[Fri05] Harvey Friedman. [FOM] Characterization of R/Simple proof, February 2005.

[LJ03] George Lakoff and Mark Johnson. *Metaphors We Live By*. University of Chicago Press, Chicago, new edition edition, April 2003.

[noa] Spintronics - Build Mechanical Circuits.

[Pav12] Dusko Pavlovic. Monoidal computer I: Basic computability by string diagrams, December 2012. arXiv:1208.5205 [cs, math].

[Pav14] Dusko Pavlovic. Monoidal computer II: Normal complexity by string diagrams, February 2014. arXiv:1402.5687 [cs, math].

[Pav23] Dusko Pavlovic. Programs as Diagrams: From Categorical Computability to Computable Categories, March 2023. arXiv:2208.03817 [cs, math].

[PY18] Dusko Pavlovic and Muzamil Yahia. Monoidal computer III: A coalgebraic view of computability and complexity, March 2018. arXiv:1704.04882 [cs, math].

[Red] Michael J. Reddy. The conduit metaphor: A case of frame conflict in our language about language. *Metaphor and Thought*, page 164.

[Ric15] Richard Ridel. Mechanical Turing Machine in Wood, October 2015.

[Wie96] Anna Wierzbicka. *Semantics : Primes and Universals*. Oxford University Press UK, Oxford England ; New York, March 1996.

[wik22] Jaquet-Droz automata, November 2022. Page Version ID: 1123627554.

[WMC21] Vincent Wang-Mascianica and Bob Coecke. Talking Space: inference from spatial linguistic meanings, September 2021. arXiv:2109.06554 [cs].