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 of iconic semantics

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

1.1 Composition of dynamic verbs via temporal anaphora

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

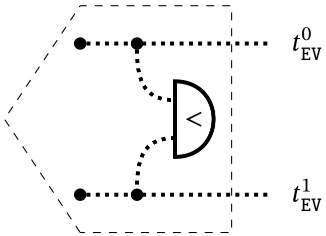
The technical difficulty I'd like to sketch a solution for is that while these parameters must be given as real numbers in the interval $[0, 1]$, temporal natural language underspecifies: e.g. in the utterance `Bob drank, and then he slept` he could have drank in the morning and then slept in the afternoon, 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 1.1.2 (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

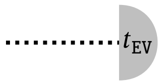
Postscript: These sketches are mostly a restructuring of content that otherwise dangled from the previous chapter. Dynamic verbs and modals are two new sketches I had in mind while initially writing the thesis but didn't make it to the submitted version. There will probably be technical errors, but the 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.

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.

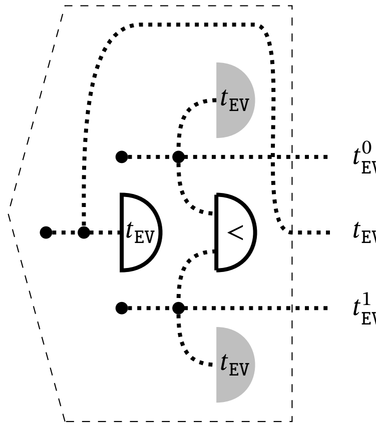
Episodic event
(Interval determined by ordered endpoints)



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

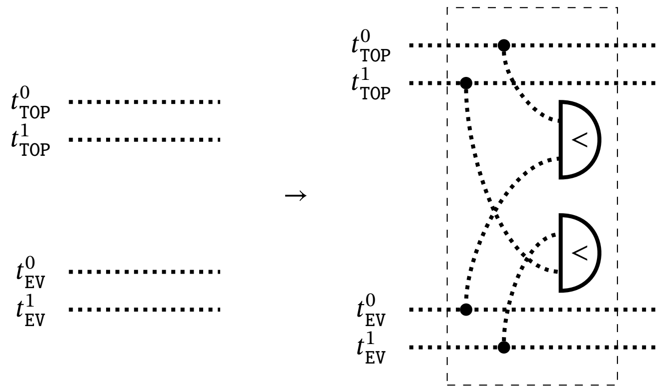


Hybrid event
(Open set with endpoints)



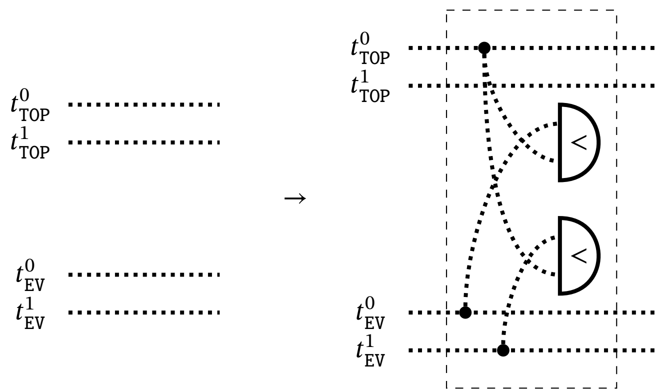
Now we model temporal aspects as circuit components — what appears to distinguish aspects from tenses is that aspects are always relative to the temporal data of two events, whereas tenses may be "intransitive" on events — so all of our aspectual data will involve constraining pairs of events (one of 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)



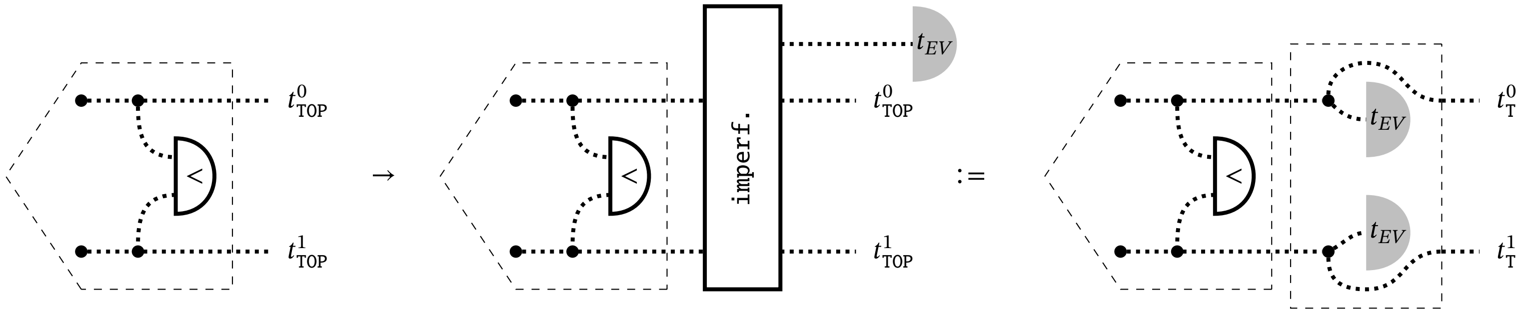
The *terminative* aspect constrains an event to occur entirely before the beginning of the topic time. Terminative composition of verbs may be glossed as `(event) and-then (topic)`, and this kind of composition yields the view of text circuits as implicitly encoding the temporal order in which gate-as-events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

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



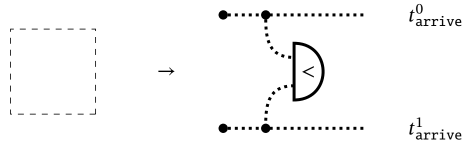
The *imperfective* aspect we consider as constraining an episodic topic time to lie within some ongoing habitual event, where the habitual event is represented as a free coparameter. In discourse, introducing an imperfective constraint corresponds to splicing in such a constraint, which we gloss as a 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)



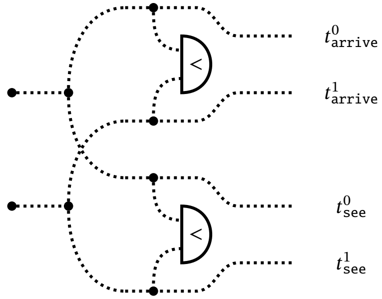
Example 1.1.3. So here is an example of Yucatan Maya taken from [CITE], which is an excerpt of an interview with a speaker fleeing a cyclone. I have split the excerpt into numbered single-verb clauses, accompanied by glosses in English with aspect-markers and the corresponding evolution of a text-circuit 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) $\left[\dfrac{\textit{Kaajk'ucho'on}\ \texttt{túun way tekàajil x Jaxleyile,}}{\texttt{When we } \underbrace{\textit{arrived}}_{\text{PERF.}}\texttt{,}}\right]$



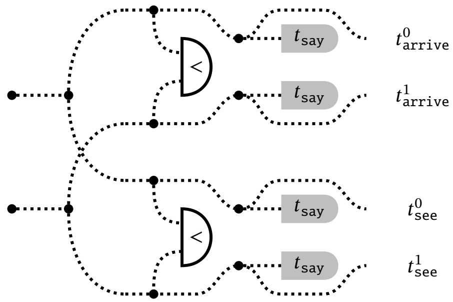
The second event is what the refugees saw, implicitly concurrent with event (1), which we opt to treat with a prepended copy of endpoints. `arrive & see` then form an atomic topic for events (3) and (4), which we deal with by constraining both (1) and (2) in the same way. Note that there is a single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

(2) [*kilike'* tuláakal máake', / we $\underbrace{\textit{saw}}_{\text{PERF.}}$,] →



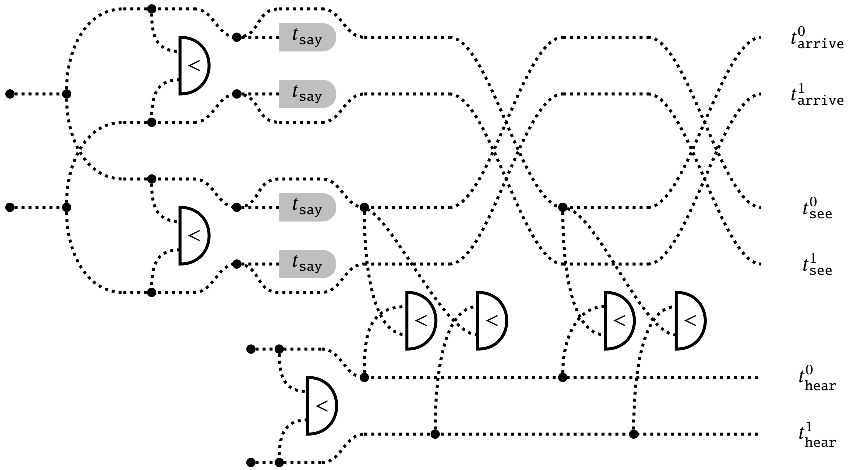
The third event refers to the villagers saying something, in the imperfective aspect with respect to events (1) and (2), so we constrain those topics accordingly. In gloss, it was an ongoing event that the villagers were saying something when the refugees arrived.

(3) [*táan uya'aliko'obe'* jach / everyone $\underbrace{\textit{was saying}}_{\text{IMPF. WRT. (1,2)}}$] →



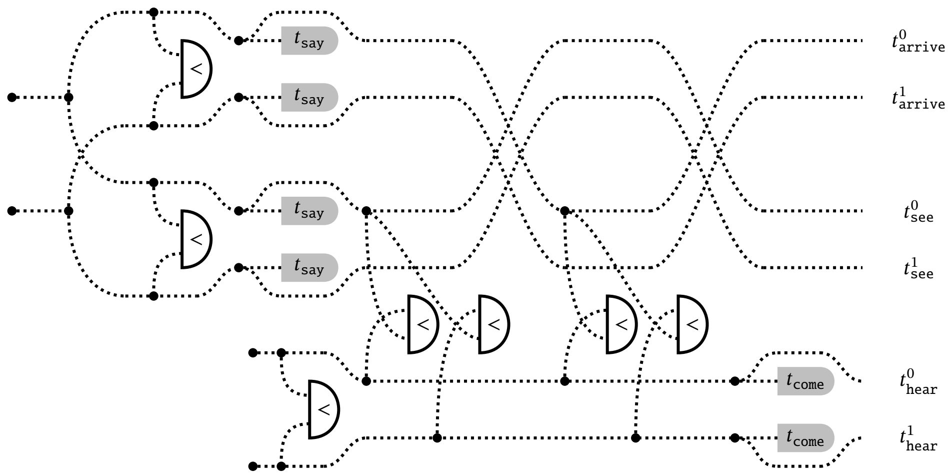
The fourth event refers to what the villagers had heard, in the terminative aspect with respect to (1) and (2). In gloss, the villagers were saying (reporting) the episodic event of them hearing something on the radio, and this hearing-event had completed before the refugees' arrival.

(4) *ts'uyu'ubiko'ob ti'* ràadyoe'
(they) *had heard* on the radio
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.*
(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.

1.2 *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 1.1: 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 1.2: 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.

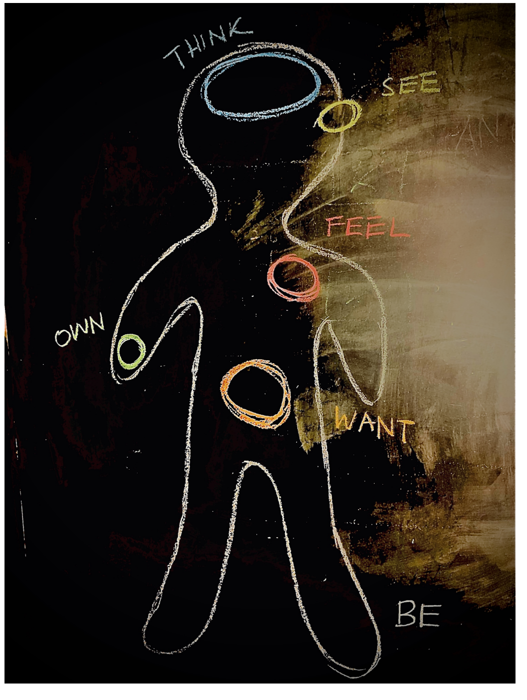


Figure 1.3: So the basic idea is to put representations of worlds inside bounded regions as containers, and in this way iconic semantics provides a univocal setting that displays all of the relevant worlds at once. We are free to pick visual conventions, as they are no more or less arbitrary than the 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.

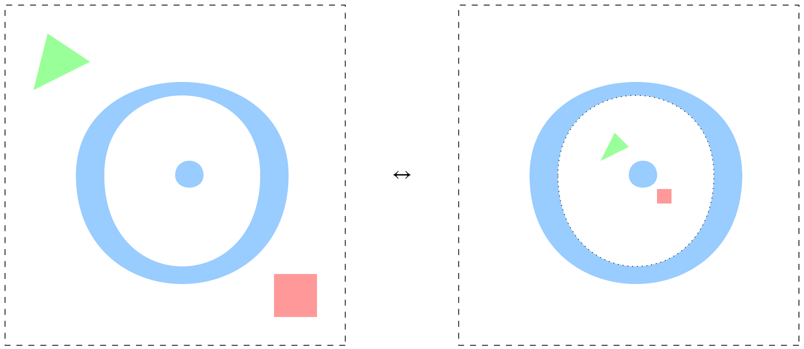


Figure 1.4: The inside and the outside of a container with a solid blob inside are both homotopic to the space with a puncture. This is only approximately a continuous bijection because the unbounded outside space can only map to the open interior of the container. We can use such bijections as a bridge 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 1.5: 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.

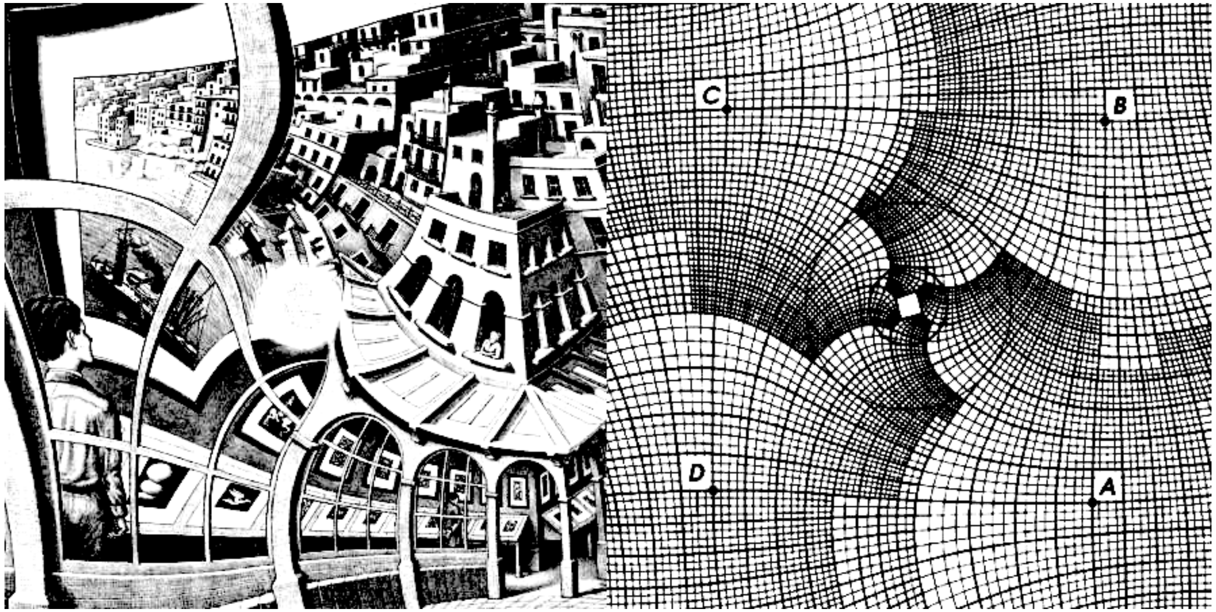


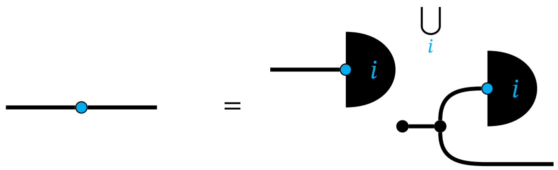
Figure 1.6: Escher's "Print Gallery" lithograph alongside his working sketch of the vortex-grid geometry the work was built on. On the left of the lithograph, an observer examines a framed painting of a town. Going clockwise, we see more details of the town, which has in it a print gallery, within which 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.

1.3 *Configuration spaces*

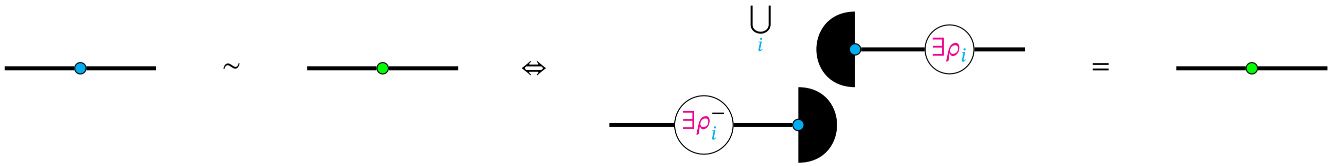
Individual sticky-spiders correspond to static collections of set-labelled shapes in **ContRel**; in this sketch I want to talk about all the different ways the same collection of shapes can be arranged in space. Let's assume for simplicity that we only want to deal with *nice* sticky-spiders where cores 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 1.3.1 (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.



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 1.3.2 (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:



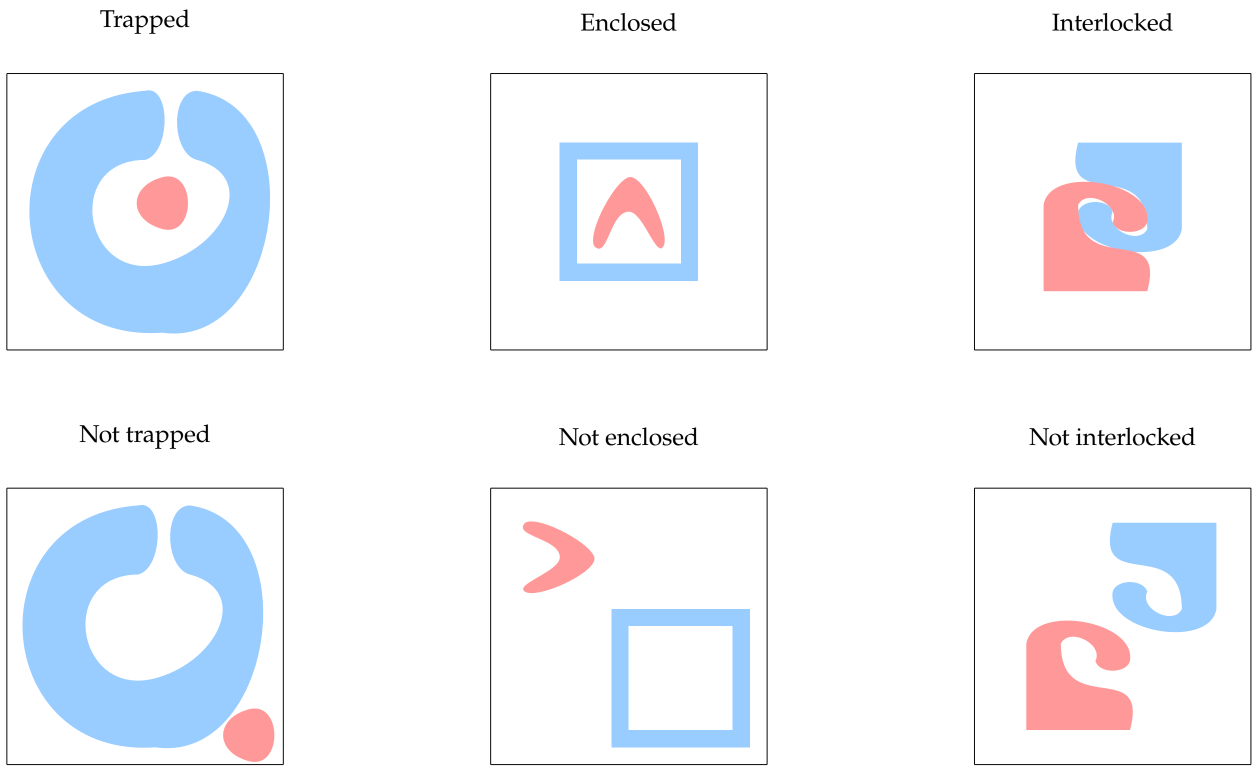
We've suppressed labelling of the states and we've contracted the cup to just depict the open state as a semi-circle.

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.

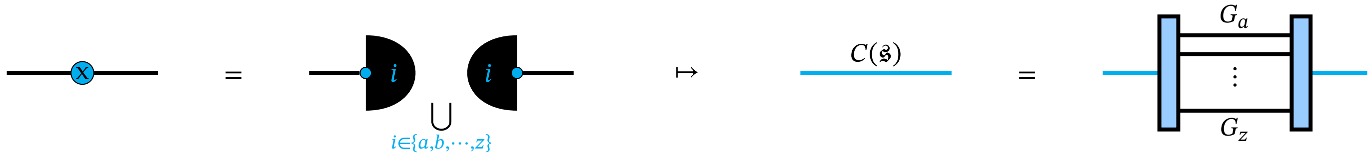
Definition 1.3.3. The *configuration space* $C(\mathfrak{s})$ of a nice spider $\mathfrak{s}$ with indexing set I is the topological space with underlying set defined to be the equivalence class $[\mathfrak{s}]$ of $\mathfrak{s}$ under displacement. Assuming the topological group of rigid displacements 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 $[\mathfrak{s}]$.

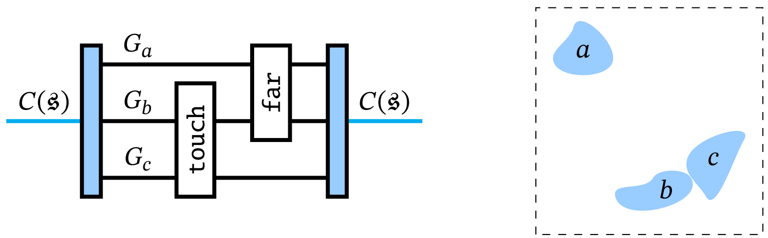
Example 1.3.4 (The connected components of configuration space)**.** Configuration space allows us to define a "slideability" relation between configurations of a spider $\mathfrak{k}$ as the endpoints of continuous functions from the unit interval into $C(\mathfrak{s})$. This in turn allows us to consider 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.



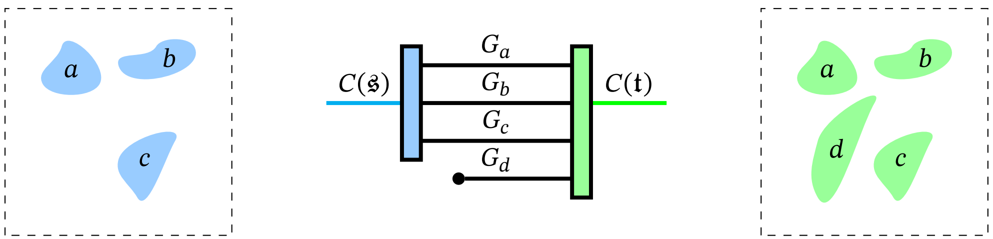
In configuration spaces we're making use of the fact that any displacement relationship comes with (up to a non-unique choice of basepoints for each component shape) a witnessing tuple of ρ_is. As a consequence, the configuration space of a sticky-spider is a retract of the product space $\bigtimes^{|I|} G$ 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 1.3.5 (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 $j \in d^{-1}(i)$, $g(j) \subseteq f(i)$.

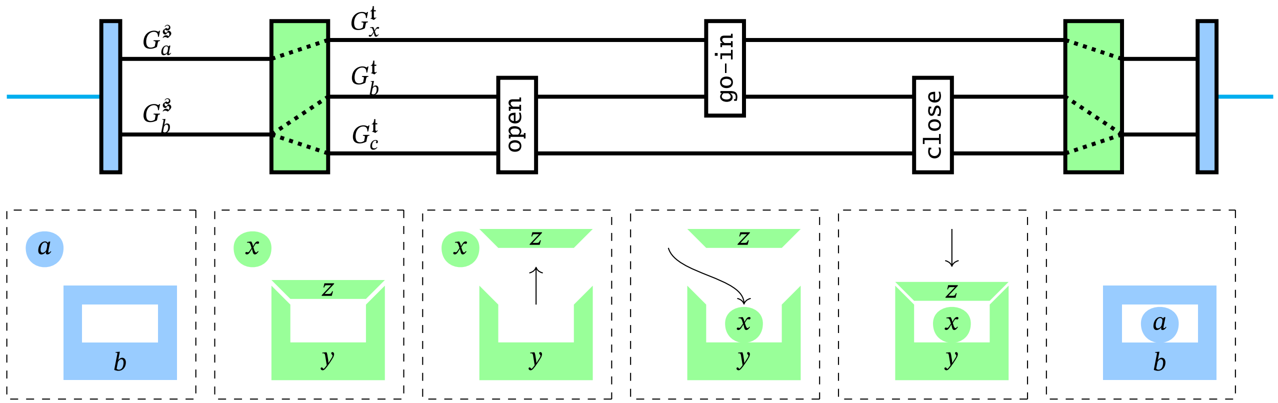
Since the composition of surjectives is also surjective and the subsethood condition is transitive, disintegration is a transitive relation. It's also reflexive, and since surjections $A \twoheadrightarrow B$ and $B \twoheadrightarrow A$ implies a bijection $A \simeq B$ and $X \subseteq Y$ with $Y \subseteq X$ 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 1.3.6 (*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.

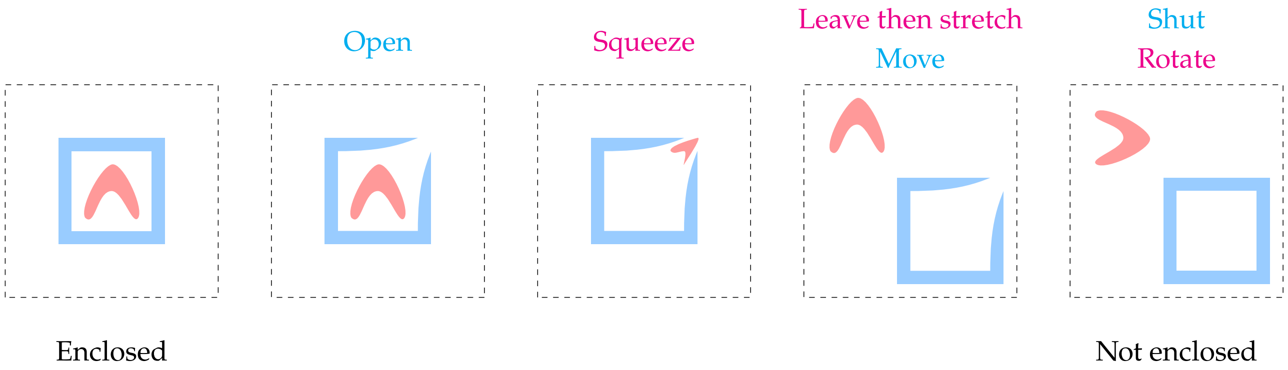
The intuition behind shattering is that the $\subseteq$-condition in the disintegration relation lets the disintegrating spider "shave a little" off of the disintegrated spider, and locally minimal disintegrations "shave the least off", doing the best they can to partition shapes. So now we get gluing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

Definition 1.3.7 (*et Coagula*)**.** $\mathfrak{t}$ is a *gluing* of $\mathfrak{s}$ if $\mathfrak{s}$ shatters $\mathfrak{t}$.

Example 1.3.8 (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.



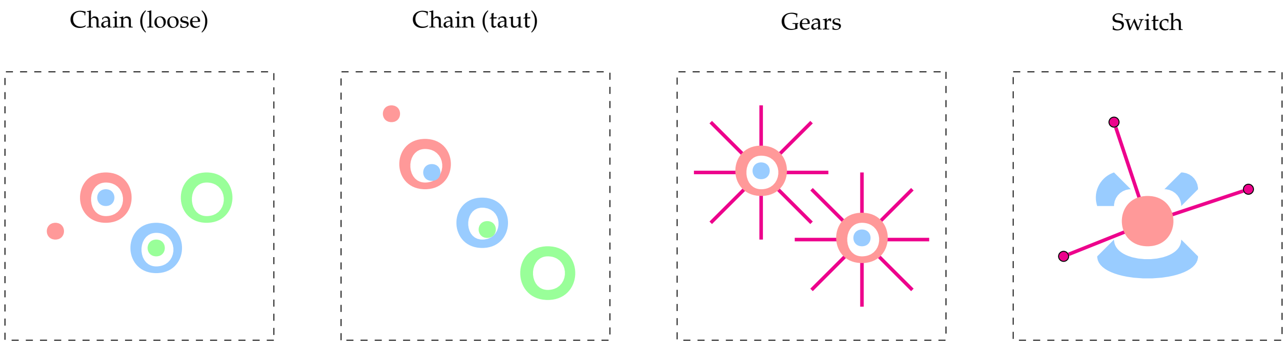
In principle, shapes can be shattered arbitrarily finely, which permits us some degree of freedom in specifying how a container opens. In conjunction with a topological group of transformations that includes scaling, we may express different ways in which things get in and out of containers, or otherwise 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 **??**. 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 1.3.9 (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:



1.4 Interpreting text circuits in **ContRel**

Recall that there were some mathematically odd choices in conventions for text circuits as syntactic objects, such as the labelling of noun wires and some subtleties with respect to interchange of parallel gates and text order. The explanation of those choices was deferred "to semantics", and since we're here now we have to make good. The aim of this section is to now explain those choices through an interpretation of text circuits in **ContRel**.

1.4.1 Sticky spiders: iconic semantics of nouns

Without loss of generality we may consider sticky spiders to be set-indexed collections of disjoint open subsets of an ambient space, i.e. a collection of shapes in space that a labelled from a set of names. So particular sticky spiders will be particular models of a collection of nouns; the indexing set names them, and the corresponding shapes in space are a particular iconic representation of those nouns in space.

1.4.2 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 1.7 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.

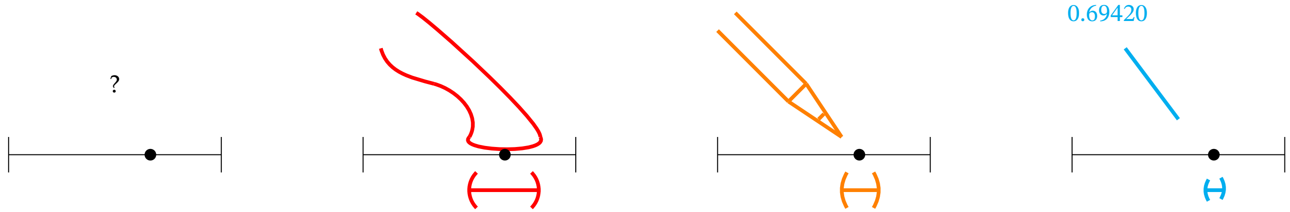


Figure 1.7: Points in space are a useful mathematical fiction. Suppose we have a point on a unit interval. Consider how we might tell someone else about where this point is. We could point at it with a pudgy appendage, or the tip of a pencil, or give some finite decimal approximation. But in each case 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

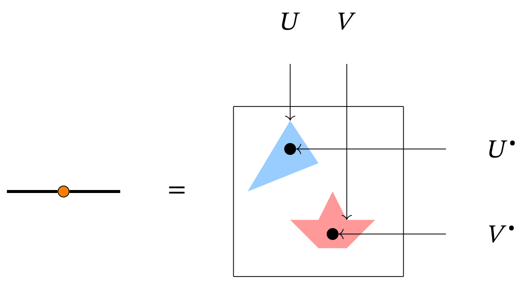
Maybe this explains the asymmetry of why tests are open sets, but why are states allowed to be arbitrary subsets? One could argue that states in this model represent what is conceived or perceived. Suppose we have an analog photograph whether in hand or in mind, and we want to remark on a particular 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 formulation of a very 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, and the states accommodate this.

1.4.3 *Configuration spaces: labelled noun wires*

Whereas sticky spiders correspond to particular iconic representations, we want to interpret text circuits in the configuration spaces of sticky spiders, so that text process-theoretically restricts, expands, and modifies a space of permissible iconic representations[1]]**Top** is symmetric monoidal closed with respect to product, why we just work there from the start? Because **Top** is cartesian monoidal, which in particular means that there is only one test (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 extremely useful representing entangled quantum states [CK17], and for reasoning about spatial relations [WMC21]. I'll briefly explain 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 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 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 state of information we can have about a qubit. So we have a case where we can know things 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

[1] [

parts, it resides in the entangled whole.. So it is via configuration spaces that we aim for an iconic semantics of general text. Recall from the definition of configuration spaces via split idempotents (Definition 1.3.3) that the section and retract diagrammatically allow the configuration space wire to open up to guitar strings we can place our circuits in. However, this section and retract pair is not uniquely determined. For example, in the case of configuration spaces of rigid transformations of shapes, one obtains a family of encodings of the configuration space of an n-shape sticky spider in the n-fold tensor of isometry spaces by choosing a base-point for each shape and rigidly transforming with respect to that basepoint, which on the plane may lead to different reifications of rotating shapes.



Interpreting text circuits with respect to a particular split idempotent of configuration space hence gives a good reason to label the noun wires: so that we can remember which space of isometries corresponds to which noun-shape in the absence of knowing how precisely the split idempotent encodes configuration space as the n-fold contribution of spatial possibilities for each noun.

1.4.4 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. When we interpret concepts as open-set tests, **ContRel** conspires in our favour by giving us free copy maps on every wire. This allows us to define a family of processes that behave like stative restrictions of possibilities.

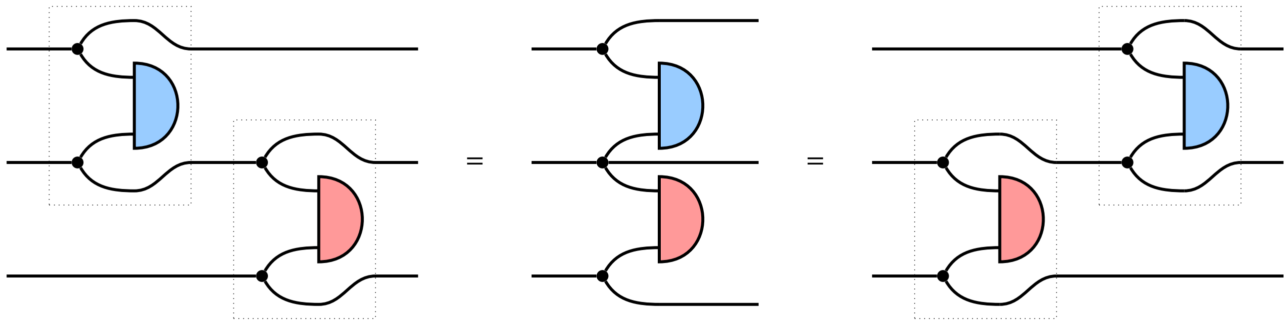


Figure 1.8:

Example 1.4.2 (Adjectives by analysis of configuration spaces). 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.

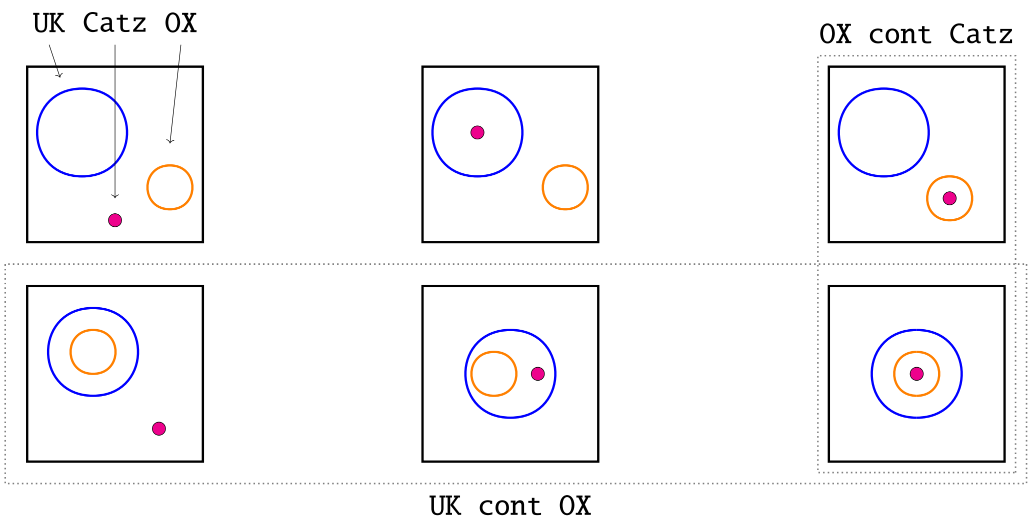
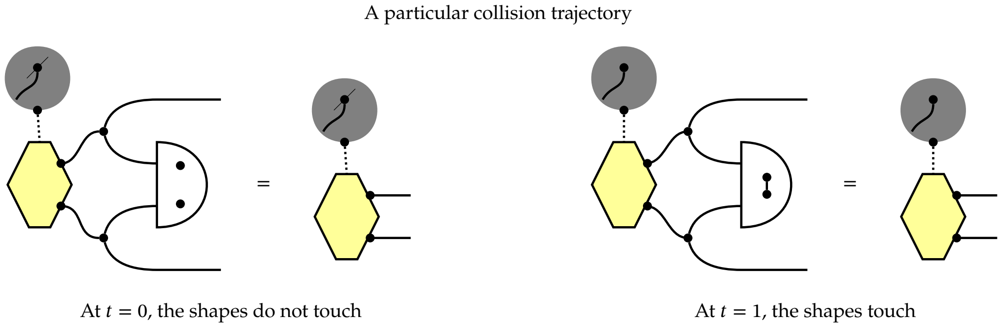


Figure 1.9: Consider the configuration space of a sticky spider on the unit square with three labelled shapes, which has 6 connected components, depicted. `Oxford contains Catz.` restricts away configurations where `Catz` is not enclosed in `Oxford`. Adding on `England contains Oxford.` further restricts 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.

1.4.5 *Homotopies: dynamic verbs and weak interchange*

Figure 1.10: *Dynamic* verbs are those that posit a change in state, such as `Bob` `goes` `home`. We want to model these verbs by homotopies, where the unit interval parameter models time. A nice and diagrammatically immediate property is that dynamic verbs are obstacles to the commutation of stative words.



Example 1.4.4 (Dynamic verbs block stative commutations). States of affairs can change after motion. A simple example is the case of *collision*, where two shapes start off not touching, and then they move rigidly towards one another to end up touching.

Figure 1.11: Recalling that homotopies between relations are the unions of homotopies between maps, we have a homotopy that is the union of all collision trajectories, which we mark ∀. We my define the interior i(∀) as the concept of collision; the expressible collection of all particular collisions. But this is not just an open set on the potential configuration of shapes, it is a collection of open sets parameterised by homotopy.

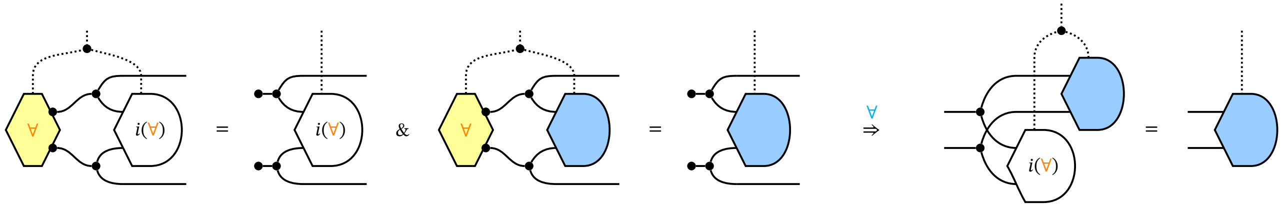


Figure 1.12: `Collision` blocks the commutation of the statives `touching` and `not touching`. In prose, the text `Alice and Bob are not touching. Alice and Bob collide. Alice and Bob are touching.` is not equal to `Alice and Bob are (touching and not touching). Alice and Bob collide.`; the latter is an incongruent state of affairs, which is reflected by an empty set of potential models (when `touching` and `not touching` intersect.)

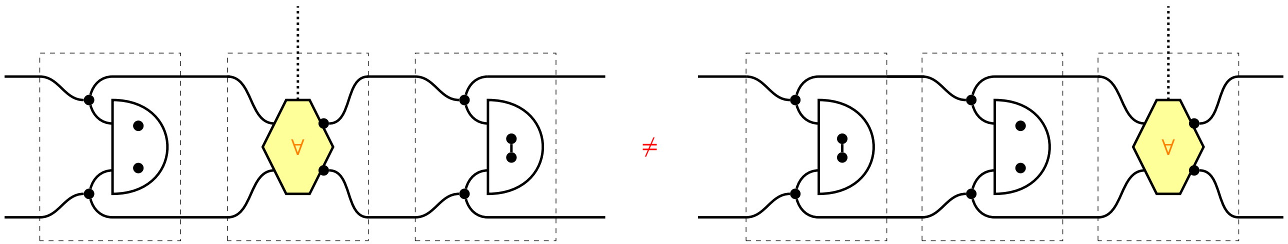
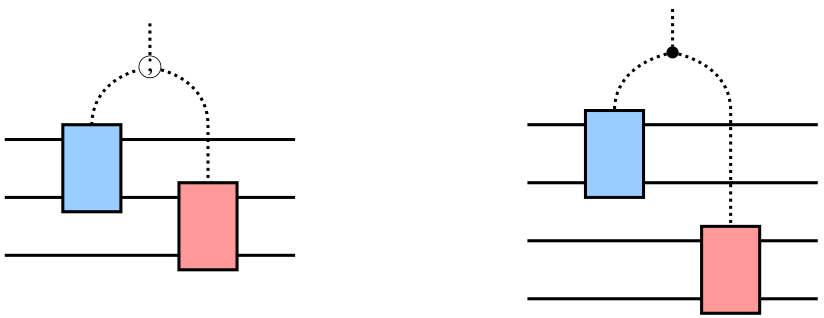


Figure 1.13: We can compose multiple motions in parallel by copying the unit interval, allowing it to parameterise multiple gates simultaneously, or compose them sequentially. The sequential composition of dynamic verbs in time explains the weak-interchange subtlety of text diagrams; there is now a diagrammatic distinction between events happening sequentially and in parallel. So now we have noncommuting gates that model *actions*, or verbs.



Sequential composition of motions

Parallel composition of motions

1.4.6 *Coclosure: adverbs and adpositions*

Figure 1.14: 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.

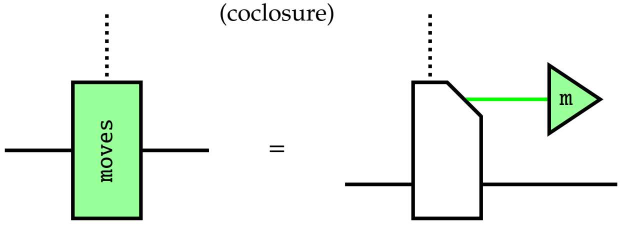


Figure 1.15: Adverb-boxes may be modelled as static restrictions in movement-space. For instance, `straight` may restrict movements to just those that satisfy some notion of path-length minimality: e.g., given a metric in movement-space on path-lengths, we may construct an open ball (Definition **??**) around the geodesic to model the adverb `straight`.

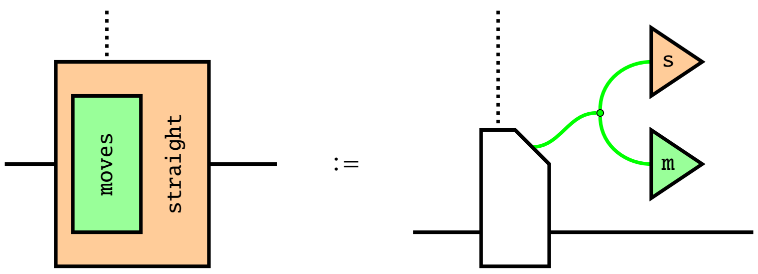
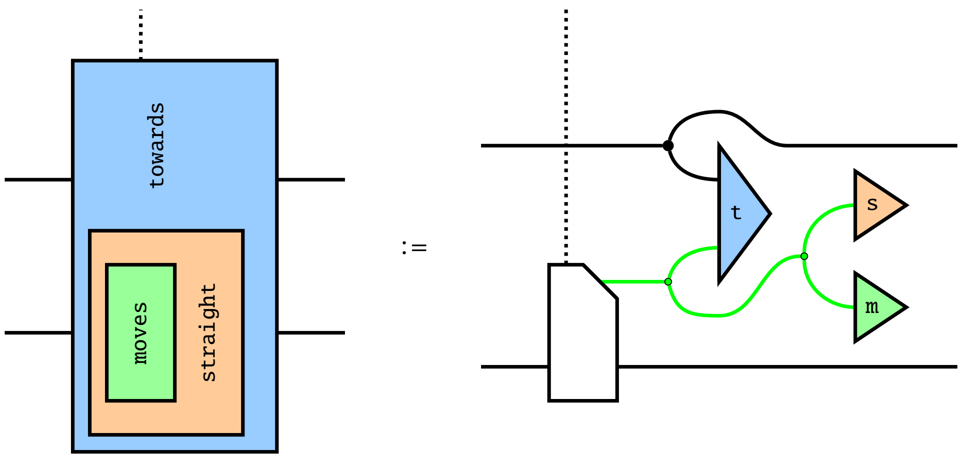


Figure 1.16: Similarly, adposition-boxes may be modelled as static restrictions on the product of the spaces of nouns and verbs. For instance, `towards` may be modelled as an open set that pairs potential positions of the thing-being-moved-towards with movements in movement-space that indeed move towards the target.



1.4.7 Turing objects: sentential complementation

Recall that we also have to explain how untyped-boxes for conjunction and sentential complementation get their semantics. One option for finite models we present here relies on an interesting property of **ContRel**, which allows **ContRel** to model untyped-boxes, as well as the linguistic phenomena of *entification* and *general anaphora*. This is because **ContRel** 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 classed *derivation*, turns words from one category into another, for example the adjective `happy` into 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.

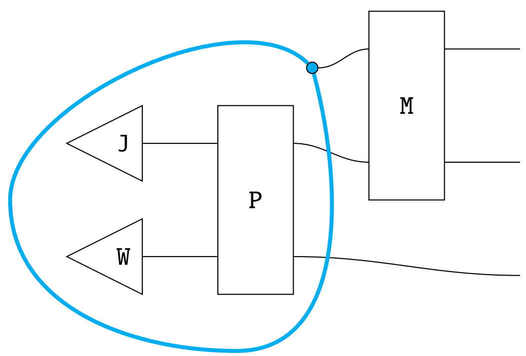


Figure 1.17:

Example 1.4.6.

`Jono was paid minimum wage but he didn't mind it`

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. I'll also call this process *entification*, which extends beyond morphology towards more complicated constructions such as a prefix `the fact that` that converts sentences into noun-like entities, insofar as these entities are valid anaphoric referents.

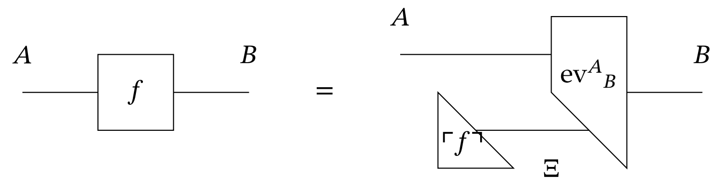
A MATHEMATICAL MODELLING PROBLEM FOR SEMANTICISTS ARISES WHEN ANYTHING CAN BE A NOUN WIRE. The problem at hand is finding the right mathematical setting to interpret and calculate with such lassos. In principle, any meaningful (possibly composite) part of text can be referred to as if it were a noun. 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. So if the code-type is just another wire in the process-theory of computation, what is its process-theoretic characterisation?

Figure 1.18:

Definition 1.4.8 (Turing object)**.** Diagrammatically, we would summarise the situation like this: for every pair of input formats and output formats (A, B), there is a computer for that format $ev^A_B : A \otimes \Xi \to B$, which takes code-format (which we will just denote Ξ going forward) as an additional input, and for every possible program $f : A \to B$, there exists a code-state $\ulcorner f \urcorner : I \to \Xi$ that evaluates to the desired program.

$$\forall A, B \in Ob(\mathcal{C})\ \exists \mathrm{ev}^A_B : A \otimes \Xi \to B\ \forall f : A \to B\ \exists \ulcorner f \urcorner : I \to \Xi$$



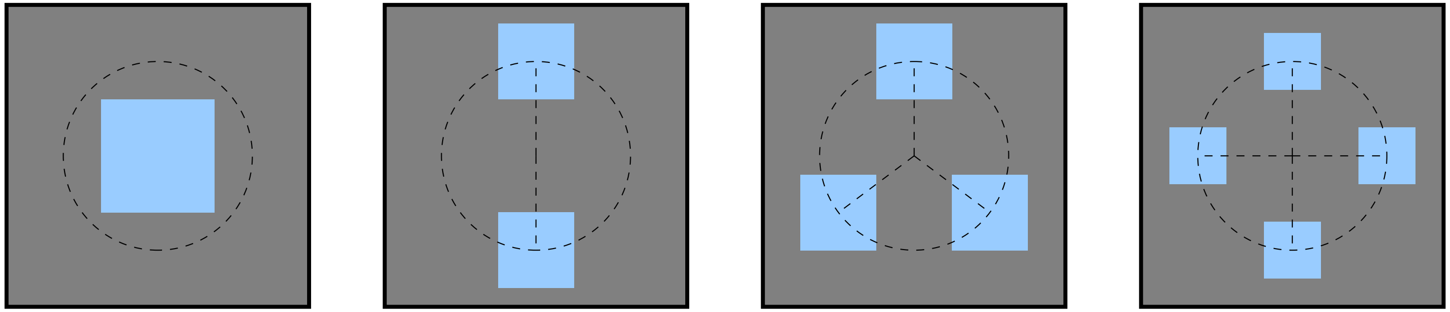
It would be true but unhelpful to conclude that any programming language is a model for text circuits, using the code data format as the noun wire. In **ContRel**, the unit square suffices.

Scholium 1.4.9. 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 code 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.

Construction 1.4.10 (Sticky spiders on the open unit square model the category relations between countable sets equipped with a code object)**.** Using the open unit square with its usual topology as the code object, there is a subcategory of **ContRel** which behaves as the category of countable sets and relations equipped with a Turing object (Construction 1.4.16.)

Proposition 1.4.11 ($(0,1) \times (0,1)$ splits through any countable set X)**.** For any countable set X, the open unit square ▦ has a sticky spider that splits through $X^\star$.

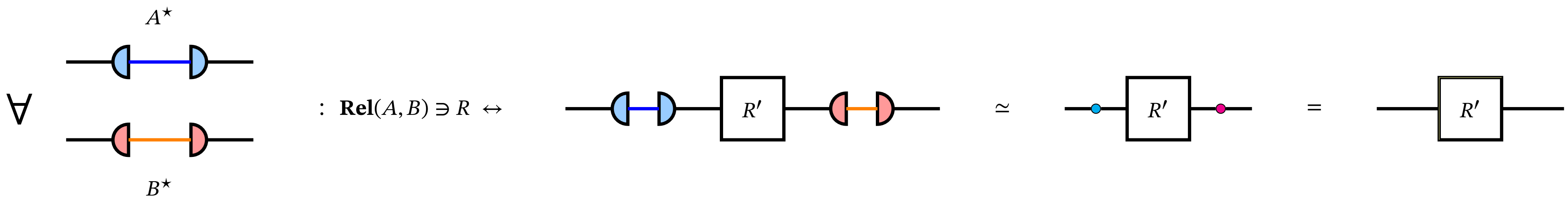
Proof. The proof is by construction. We'll assume the sticky-spiders to be mereologies, so that cores and halos agree. Then we only have to highlight the copiable open sets. Take some circle and place axis-aligned open squares evenly along them, one for each element of X. The centres of the open squares lie on the circumference of the circle, and we may shrink each square as needed to fit all of them.



□

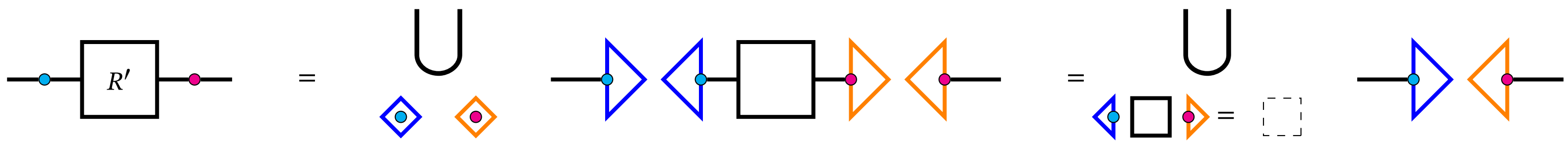
Definition 1.4.12 (Morphism of sticky spiders)**.** A morphism between sticky spiders is any morphism that satisfies the following equation.

Proposition 1.4.13 (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 $R : A \to B$.



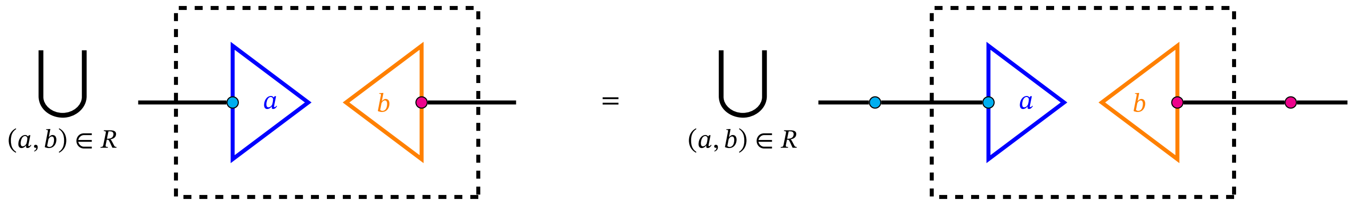
Proof.

($\Leftarrow$) : Every morphism of sticky spiders corresponds to a relation between sets.



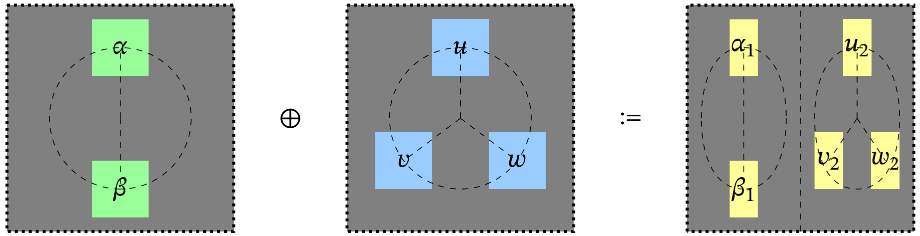
Since (co)copiables are distinct, we may uniquely reindex as:

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



$\square$

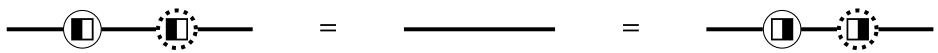
Construction 1.4.14 (Representing sets in their various guises within ▦)**.** We can represent the direct sum of two ▦-representations of sets as follows.



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.

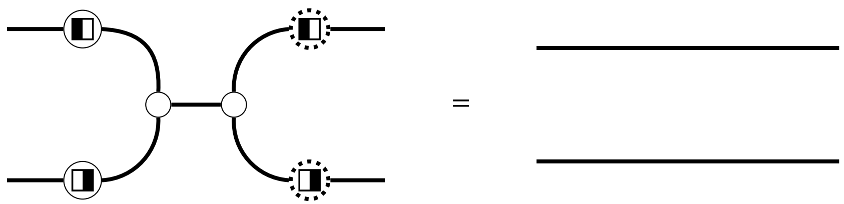
$(x, y) \mapsto (\frac{x}{2}, y)$ $\quad$ $(x, y) \mapsto (\frac{x+1}{2}, y)$ $\quad$ $(x, y)|_{x<\frac{1}{2}} \mapsto (2x, y)$ $\quad$ $(x, y)|_{x>\frac{1}{2}} \mapsto (2x - 1, y)$

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

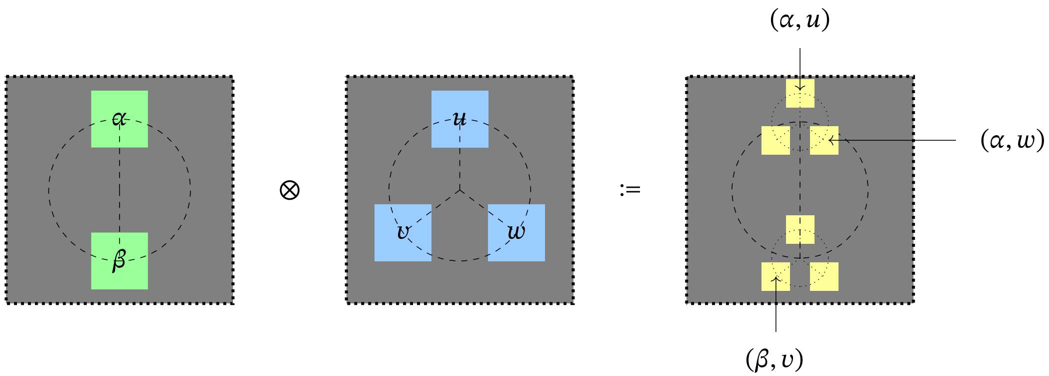


Now, to put the two halves together and to take them apart, we introduce the following two relations. In tandem with the squishing and stretching we have defined, these will behave just as the projections and injections for the direct-sum biproduct in **Rel**.

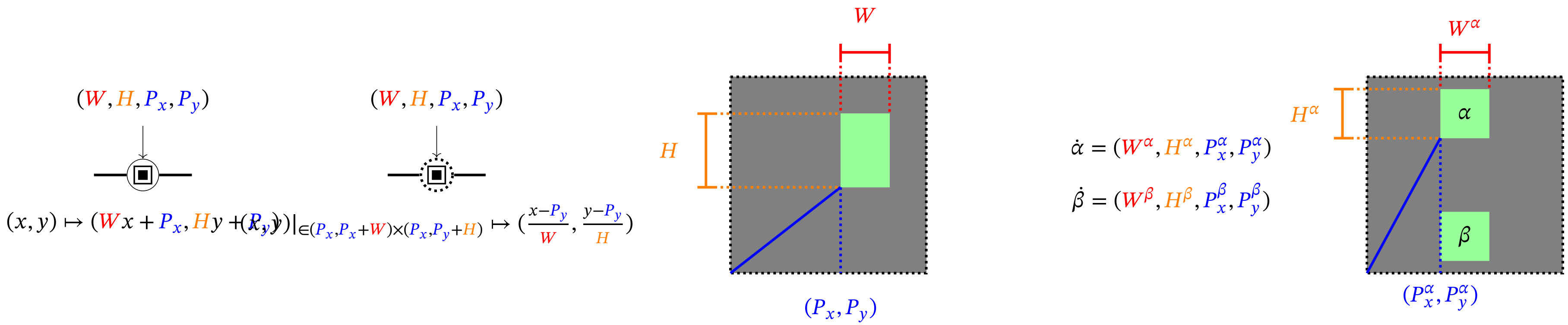
The following equation tells us that we can take any two representations in ▦, put them into a single copy of ▦, and take them out again.



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

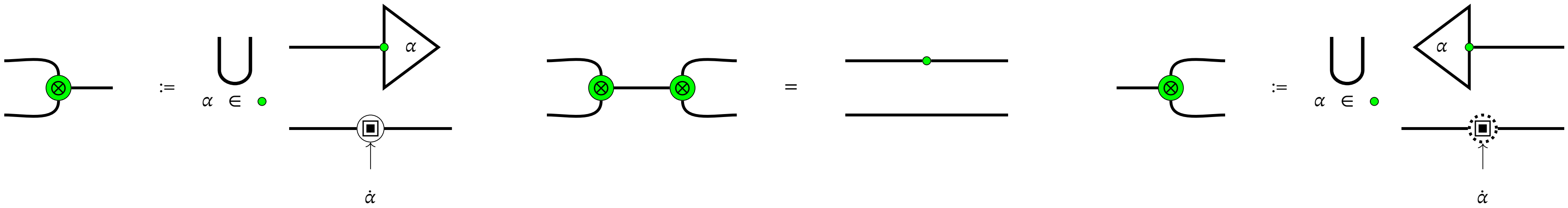


The important bit of technology here is a family of homeomorphisms of ▦ parameterised by axis-aligned open boxes, that allow us to squish and stretch spaces. Thus for every representation of a set in ▦ by a sticky-spider, where each element corresponds to an axis-aligned open box, we can associate each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



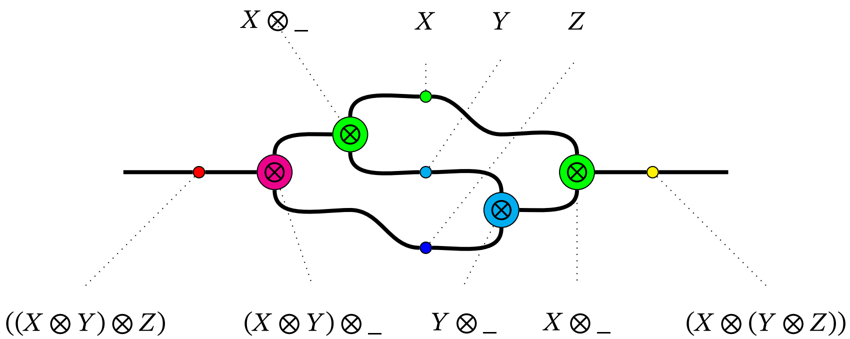
Remark 1.4.15. The essential idea is that the whole of the unit square is homeomorphic to part of it. In particular this means (modulo a point), we may make copies of shapes outside a container that are in homeomorphic correspondence with shapes within a container, the classic example being thought bubbles 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 1.4.16 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.

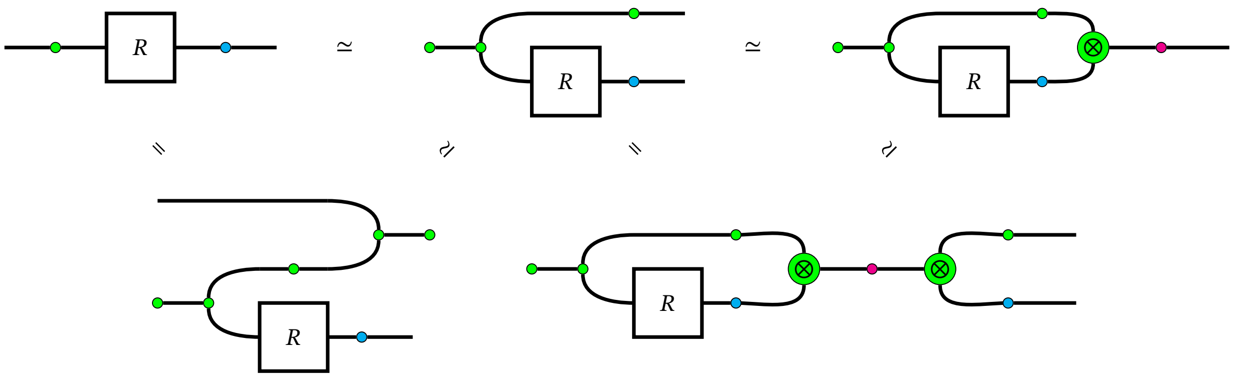


The tensor and cotensor behave as we expect from proof nets for monoidal categories.

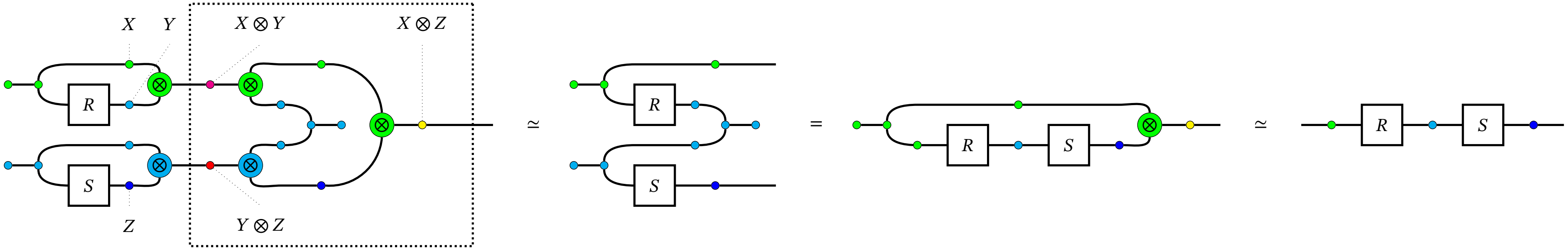
And by construction, the (co)tensors and (co)pluses interact as we expect, and they come with all the natural isomorphisms between representations we expect. For example, below we exhibit an explicit associator natural isomorphism between representations.



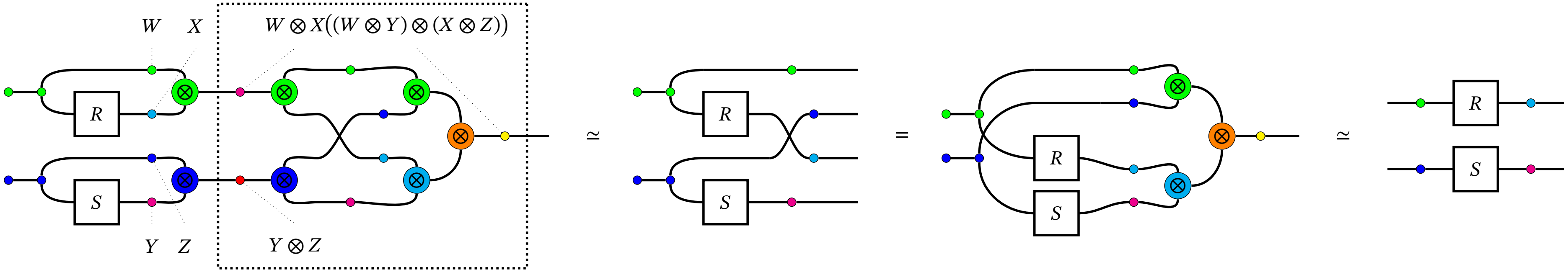
Construction 1.4.16 (Representing relations between sets and their composition within ▦)**.** With all the above, we can establish a special kind of process-state duality; relations as processes are isomorphic to states of ▦, up to the representation scheme we have chosen. This is part of the condition 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.



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].

[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].

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

[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].