



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

1	Continuous relations for semantics	5	
1.1	Topological concepts in flatland via ContRel	6	
1.2	Interpreting text circuits in ContRel	33	
1.2.1	Sticky spiders: iconic semantics of nouns	33	
1.2.2	Open sets: concepts	33	
1.2.3	Configuration spaces: labelled noun wires	34	
1.2.4	Copy: stative verbs and adjectives	35	
1.2.5	Homotopies: dynamic verbs and weak interchange	36	
1.2.6	Coclosure: adverbs and adpositions	38	
1.2.7	Turing objects: sentential complementation	39	
2	Bibliography	47	(Acknowledgements will go in a margin note here.)

1

Continuous relations for semantics

We want to reason formally with and about pictorial iconic representations, of the sort one might draw to solve a problem in elementary geometry stated in words, involving topological concepts such as **touching** and **inside**. To do this in string diagrams, I introduce and investigate the category of continuous relations, **ContRel**.

1.1 Topological concepts in flatland via *ContRel*

The goal now is to demonstrate the use of the sticky spiders we have obtained in the previous section as formal semantics for the kinds of schematic doodles or cartoons we would like to draw. Throughout we consider sticky spiders on \mathbb{R}^2 . In service of defining *configuration spaces* of shapes up to rigid displacement, we diagrammatically characterise the topological subgroup of isometries of \mathbb{R}^2 by building up in the diagrammatic presentations of the unit interval, metrics, and topological groups. To further isolate rigid displacements that arise from continuous sliding motion of shapes in the plane (thus excluding displacements that result in mirror-images), we diagrammatically characterise an analogue of homotopy in the relational setting. Finally, in we build up a stock of topological concepts and study by examples how implementing these concepts within text circuits explains some idiosyncrasies of the theory: namely why noun wires are labelled by their noun, why adjective gates ought to commute, and why verb gates do not.

ITINERARY OF UPCOMING EXAMPLES AND DEFINITIONS: I would like to say it's obvious how to make shapes move and tell when they are touching and so on, but you may not believe me. So, I am going to recover a fair chunk of elementary topology diagrammatically in *ContRel*.

SHAPES AND PLACES We can already do interesting things with just sticky spiders. In Example 1.1.5, we show how we can use sticky spiders to tell how pieces are on a chessboard. How is it that pieces on a chessboard in *continuous* space are translated into *discrete* data, such as `pawn on e4`?

THE REALS To begin modelling more complex concepts, we first need to extend our topological tools. The first stop is the reals. 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 1.1.1 (Friedman). Let $((X, \tau), <)$ be a topological space with a total order. If there exists a continuous map $f : X \times X \rightarrow X$ such that $\forall a, b \in X : a < f(a, b) < b$, then X is homeomorphic to \mathbb{R} . [Fri05]

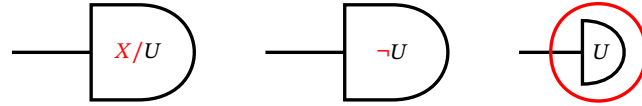
We diagrammatically characterise total orders in Definition 1.1.6. Here Trichotomy requires us to appeal to the rig structure so that we can access unions of processes explicitly. Going forward we will also introduce quantifiers into process-theoretic equations, essentially treating process-equations as we would any other symbolic algebraic specification. Then we express Friedman's theorem using diagrammatic equations in Definition 1.1.7.

THE UNIT INTERVAL If we have the unit interval, we can begin to define what it would mean for spaces to be connected (by drawing lines between points in those spaces), and we can also move towards defining motion as movement along a line. To obtain the unit interval (up to homeomorphism) we need to add endpoints to

\mathbb{R} . We can introduce endpoints for open intervals directly by asking for the space X to have points that are less than or greater than all other points. Another method, which we will use here for primarily aesthetic reasons, is to use endocombinators to define suprema. For a motivating example of the ubiquity of endocombinators, consider the case when we have a locally indiscrete topology:

Definition 1.1.2 (Locally indiscrete topology). (X, τ) is *locally indiscrete* when every open set is also closed.

If we know that a topology is locally indiscrete and we are given an open U , we would like to notate the complement X/U – which we know to be open – as any of the following, which only differ up to notation.



Unfortunately, the complementation operation $X/-$ is not in general a continuous relation, hence in the latter-most expression above we resort to using bubbles – endocombinators – as syntactic sugar. Endocombinators are like functional expressions applied to diagrams: the semantics and notation for which we borrow and modify from [HS20].

Definition 1.1.3 (Partial endocombinator). In a category \mathcal{C} , a *partial endocombinator* on a homset $(C)(A, B)$ is a function $(C)(A, B) \rightarrow (C)(A, B)$

SIMPLY CONNECTED SHAPES AND SPACES

Once we have a unit interval, we can define the usual topological notion of a simply connected space (Definition 1.1.11): one where any two points can be connected by a continuous line without leaving the space.

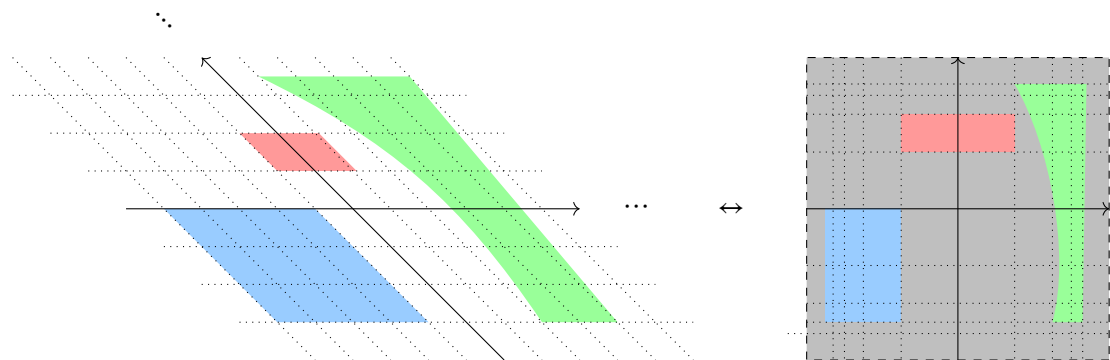
DISPLACING SHAPES

Static shapes in space are nice, but moving them around would be nicer. So we have to define a stock of concepts to express rigid motion. Rigidity however is a difficult concept to express in topological spaces up to homeomorphism – everyone is well aware of the popular gloss of topology in terms of coffee cups being homeomorphic to donuts. To obtain rigid transformations as we have in Euclidean space, we need to define metrics (Definition 1.1.13), and in order to do that, we need addition (Definition 1.1.12).

RIGID DISPLACEMENTS

Then we return to sticky spiders and we restrict our consideration to those on the open unit square, so that we can speak of shapes on a canvas. We want to rigidly displace the shapes of a sticky spider, which is obtained via the notions of isometries (Definition 1.1.16) and topological groups (Definition 1.1.15).

Remark 1.1.4. When we draw on a finite canvas representing all of euclidean space, properly there should be a fishbowl effect that relatively magnifies shapes close to the origin and shrinks those at the periphery, but that is only an artefact of representing all of euclidean space on a finite canvas. Since all the usual metrics are still really there, going forward we will ignore this fishbowl effect and just doodle shapes as we see fit.



MOTION OF RIGID BODIES

To obtain continuous transformations in the plane from the configuration of shapes in one spider to end at the configuration of shapes in another, we define a conservative generalisation of *homotopy*: the continuous deformation of one map to another (Definition 1.1.20). A side benefit we obtain is that when we have homotopies, we can also define contractible shapes, which are connected and have no holes. Then we define configuration spaces of shapes on the plane so we can talk about all permissible positions shapes may be in, and at last we can define motions of rigid bodies.

MODELLING LINGUISTIC TOPOLOGICAL CONCEPTS

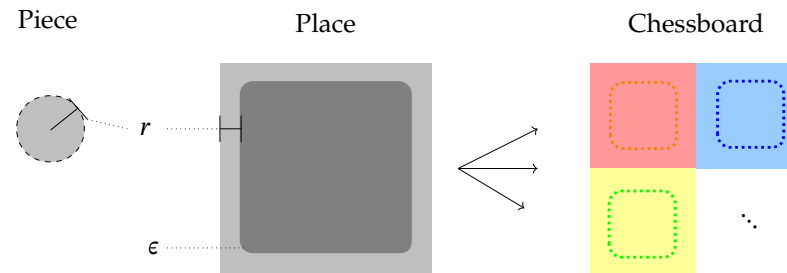
By "linguistic", I mean to refer to the kinds of concepts we use in everyday language, and by "topological" I *don't* mean spatial relationships that are invariant under homeomorphism (as recall, we deal primarily with rigid and finitely deformable objects in our day-to-day). I mean to refer to non-metric and non-directional spatial relationships such as parthood, touching, within/surrounds, containment/entrapment, interlocking, and mutually constrained motions. These are concepts that even very young children have an intuitive grasp of [Jea67], but their formal definitions are difficult to pin down. One such relation modelled here – touching – is in fact a *semantic prime* [Wie96]: a word that is present in essentially all natural languages that is conceptually primitive, in the sense that it resists definition in simpler terms. It is among the ranks of concepts like *wanting* or *living*, words that are understood by the experience of being human, rather than by school. As such, I make no claim that these definitions are canonical, just that they are good enough to reason with. Then I will introduce some dynamic relationships such as collision, along with some remarks on how vignettes may be composed out of homotopies and a general recipe for obtaining homotopies that – via

deformation – relate any pair of sticky spiders with the same number of shapes. This analysis extends to talk of rigid and deforming bodies and the manner, order, and coordination of their movement and interaction in any space.

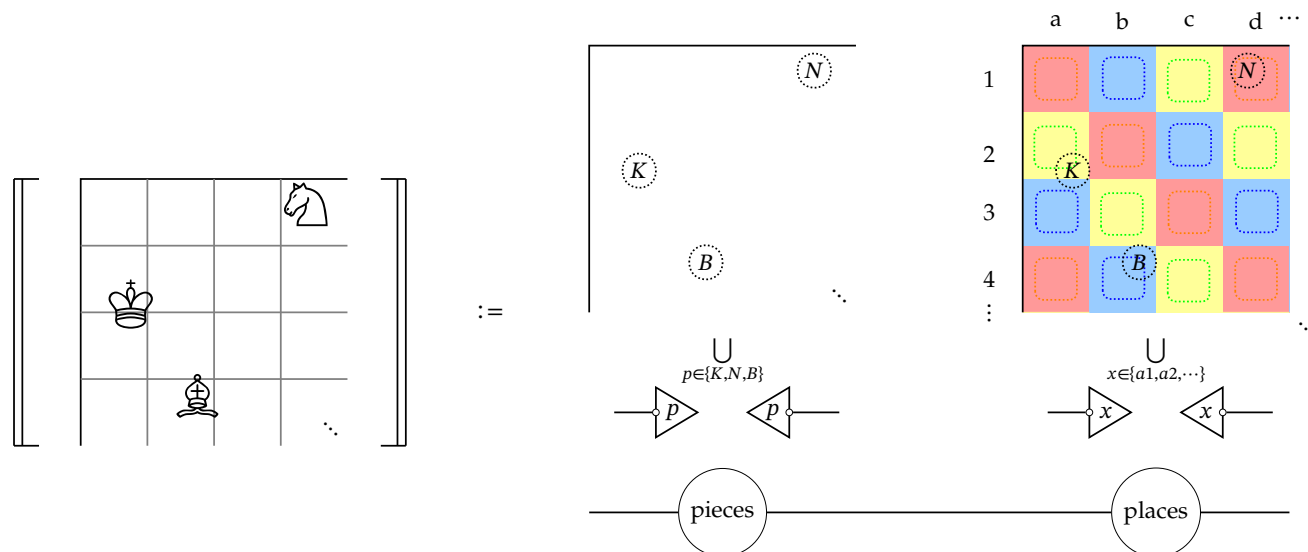
THUS, I CONSIDER ANY LINGUISTIC SEMANTICS THAT CAN BE GROUNDED IN MECHANICAL OR TABLETOP MODELS, TO BE FORMAL. By Example 1.1.5, we have enough technology to speak of locations in space, so we have access to "tabletop semantics": anything that in principle can be represented by counters and meeples in a boardgame, with for instance reserved spaces on the board for health and hunger and whatever else is necessary. The analysis of rigid bodies is hopefully sufficient to demonstrate that, in principle, one may linguistically characterise mechanical models, and there are a lot of mechanical models, including: mechanically realised clocks [duh.], analogues of electric circuits [noa], computers [Ric15], and human-like automata [wik22]. Of course in reality mechanical motions are reversible among rigid objects, and directional behaviour is provided by a source of energy, such as gravitational potential, or wound springs. But we may in principle replace these sources of energy by a belt that we choose to spin in one direction – our own arrow of time.

OBJECTION: THAT IS 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 ?? . Part of the reason I went through the trouble of deriving linguistic topological relations from nothing is so that I can claim that by any sensible definition, I am doing formal semantics for natural language. Whether or not I'm exceeding the scope of what a linguist might consider formal semantics is ultimately irrelevant, as I am not concerned with the modal human mechanism, only with process-theoretic characterisations of things, because they tend to be worthwhile practically. There is maybe also a prejudice that formal semantics must necessarily resolve in some symbolic logic, to which I might charitably respond that I'm working with an algebraic system, just not a one-dimensional one. Less charitably, I don't care what you think linguistics ought to be.

Example 1.1.5 (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. As a technical aside, to keep the core of the tiles open, we can choose an arbitrarily sharp curvature ϵ at the corners.



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.

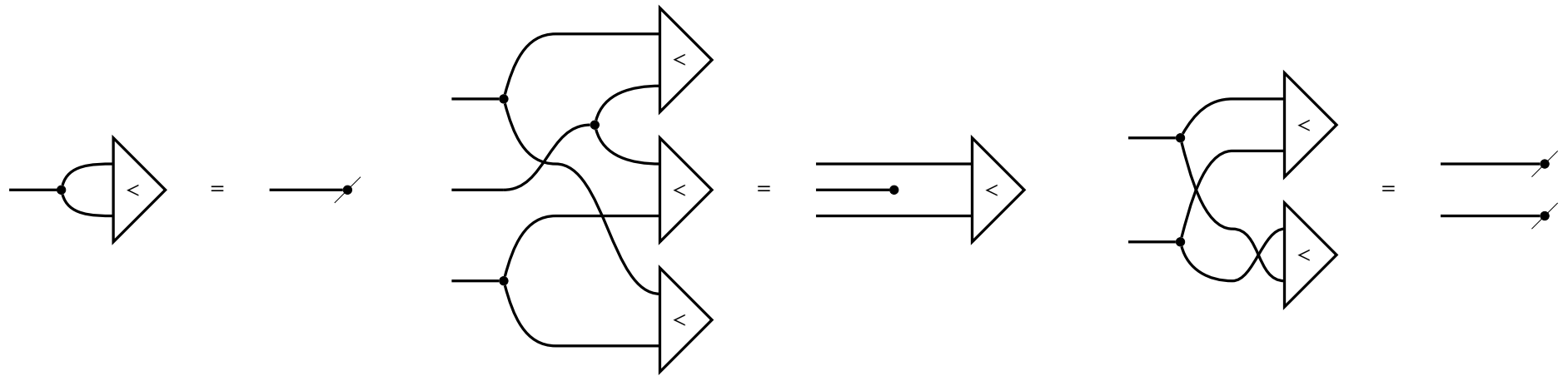


Definition 1.1.6 (Less than). We define a total ordering relation $<$ as an open set on $X \times X$ that obeys the usual axiomatic rules:

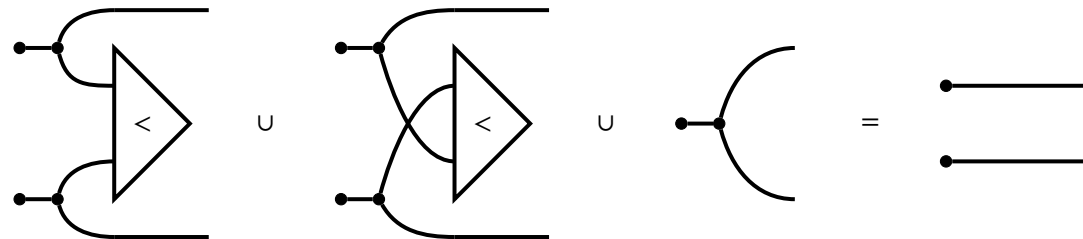
Antireflexive ($\forall x : x \not< x$)

Transitive ($\forall xyz : x < y \ \& \ y < z \Rightarrow x < z$)

Antireflexive $\forall xy \neg(x < y \ \& \ y < x)$

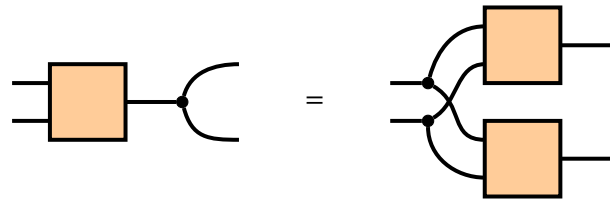


Trichotomy ($\forall xy : x < y \vee y < x \vee x = y$)

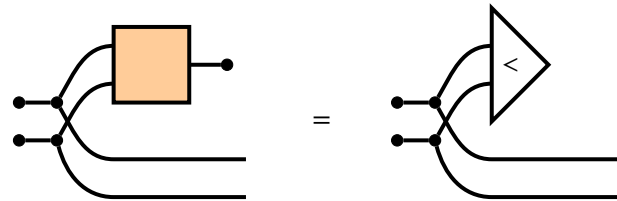


Definition 1.1.7 (Friedman's function). Just as a wire in **ContRel** has the discrete topology if it possesses spider structure (Proposition ??), a wire is homeomorphic to the real line by Theorem 1.1.1 if it possesses an open that behaves as Definition 1.1.6, and a map that satisfies:

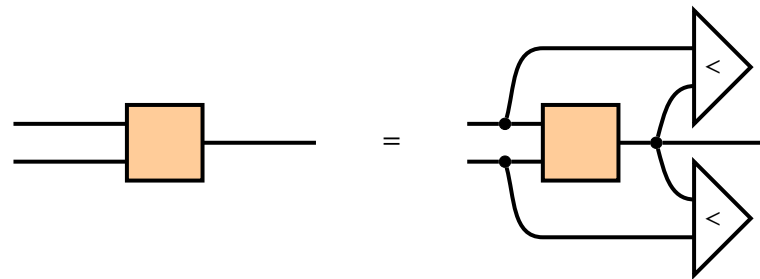
(Continuous) partial function...



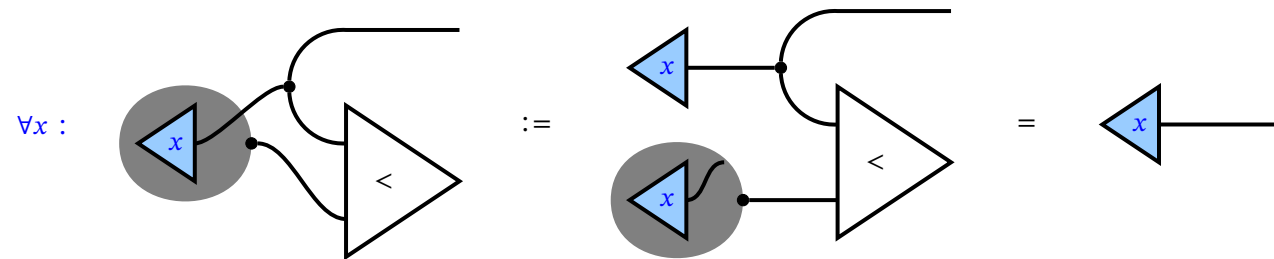
...defined on domain $(a, b) : a < b$



...such that $a < f(a, b) < b$

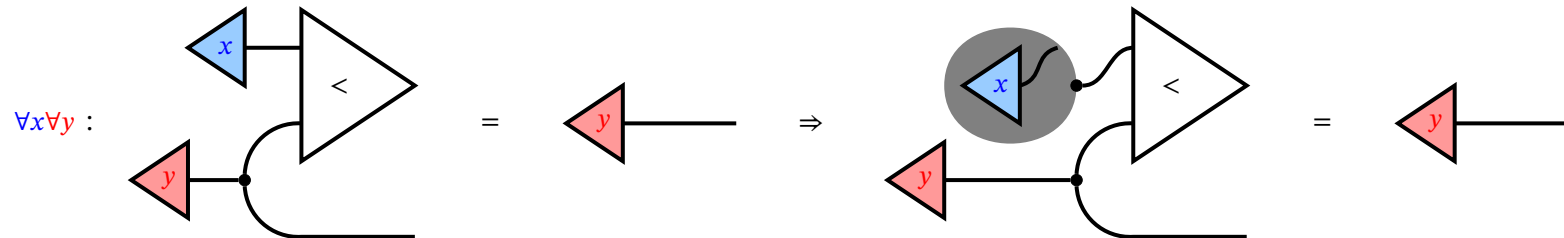


Definition 1.1.8 (Upper and lower bounds via endocombinators). Upper bounds are endocombinators that send states to points. This is an equational condition quantified over all states.



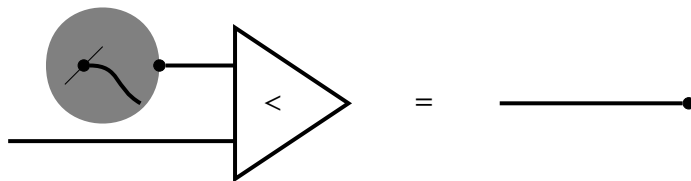
We can add in further equations governing the upper bound endocombinator to turn it into a supremum, where the lower endpoint is obtained as the supremum of the empty set, and the upper endpoint is the supremum of the whole set.

Definition 1.1.9 (Suprema).

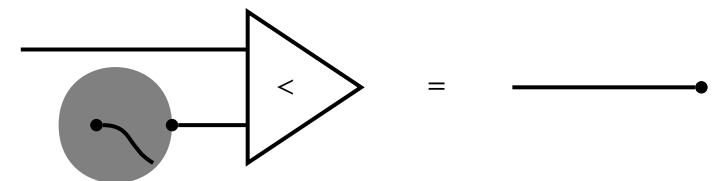


Definition 1.1.10 (Endpoints). Now we can define endpoints purely graphically. The lower endpoint is the supremum of the empty state, and the upper the supremum of everything.

Lower endpoint (0)



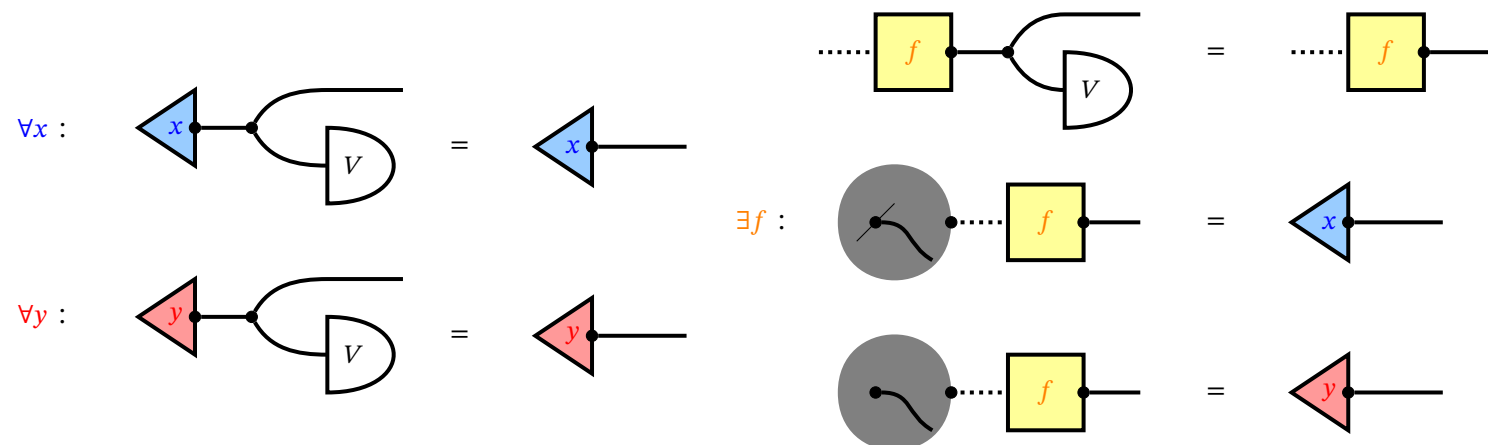
Upper endpoint (1)



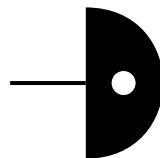
Going forward, we will denote the unit interval using a thick dotted wire.

Definition 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 .

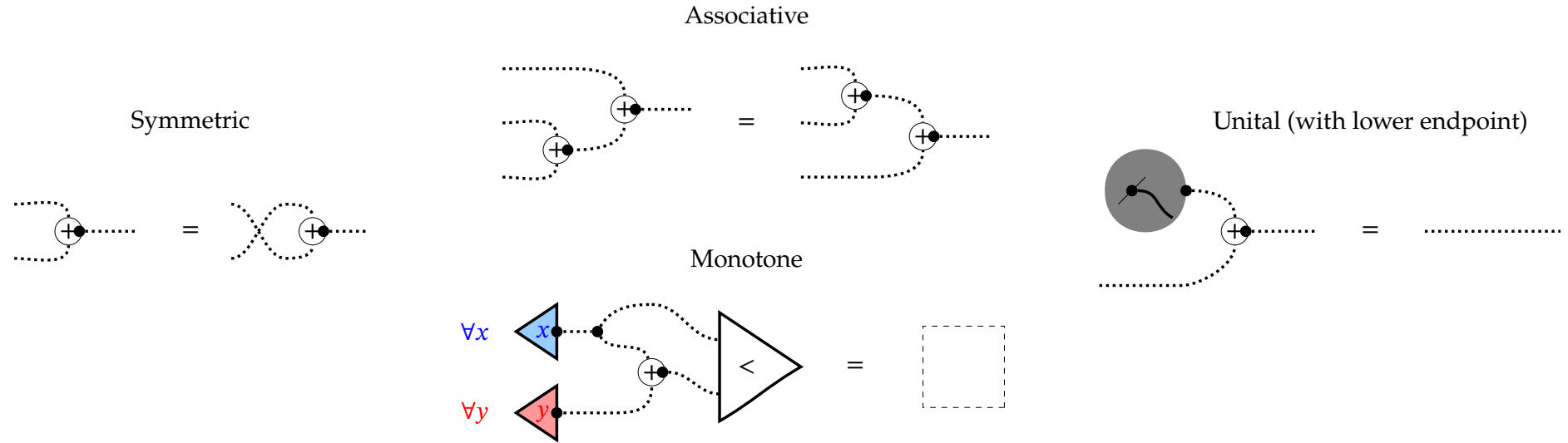
V is simply connected when:



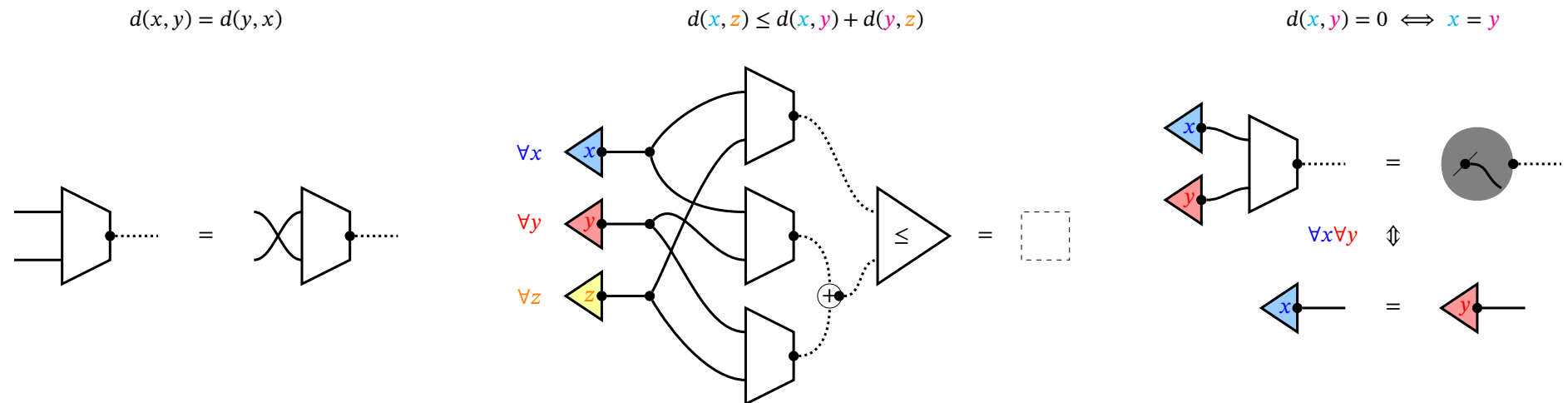
Simple connectivity is a useful enough concept that we will notate simply connected open sets as follows, where the hole is a reminder that simply connected spaces might still have holes in them.



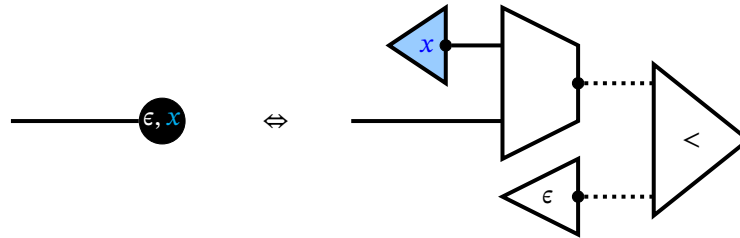
Definition 1.1.12 (Addition). In order to define metrics, we must have additive structure, which we encode as an additive monoid that is a function. All we need to know is that the lower endpoint of the unit interval stands in for "zero distance" – as the unit of the monoid – and that adding positive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



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



Definition 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.

With metrics in hand, we can proceed to define isometries as topological groups that keep distances invariant.

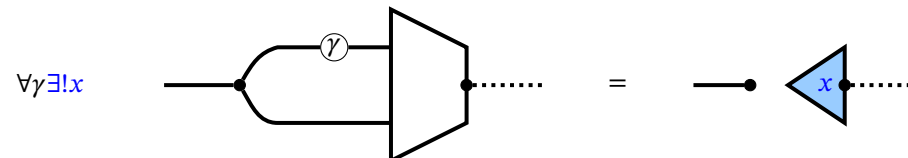
Definition 1.1.15 (Topological groups). It is no trouble to depict collections of invertible transformations of spaces $X \rightarrow X$.

A topological group G

$$\forall \gamma \in G \exists \gamma_{\in G}^{-} \quad \text{---} \gamma \text{---} \gamma^{-} \text{---} = \text{---}$$

A consequence of invertibility and the requirement that the identity transform is a group element forces all transformations in a topological group to be functions.

Definition 1.1.16 (Isometry). An isometry is a distance preserving transformation: applying the metric pointwise before and after a topological group action gives the same value.



Definition 1.1.17 (Planar isometries). We know the planar isometries of Euclidean space can be expressed as a translation, rotation, and a bit to indicate the chirality of the shape – as mirror reflections are also an isometry. In flatland, we do not expect shapes to suddenly flip over. We would like to express just those rigid transformations that leave the chirality of the shape intact, because really we want to only be able to slide the shapes around the canvas, not leave the canvas to flip over.

Isometries of \mathbf{R}^2

$$\begin{array}{c}
 \text{---} \circ \gamma \text{---} = \text{---} \circ (\mathbf{x}, \theta, c) \text{---} \\
 \mathbf{x} \in \mathbf{R}^2 \\
 \theta \in S^1 \simeq [0, 2\pi) \\
 c \in \{-1, 1\}
 \end{array}$$

With this in mind, we have the following condition relating different spiders on the unit square, telling us when one is the same as the other up to rigidly displacing shapes.

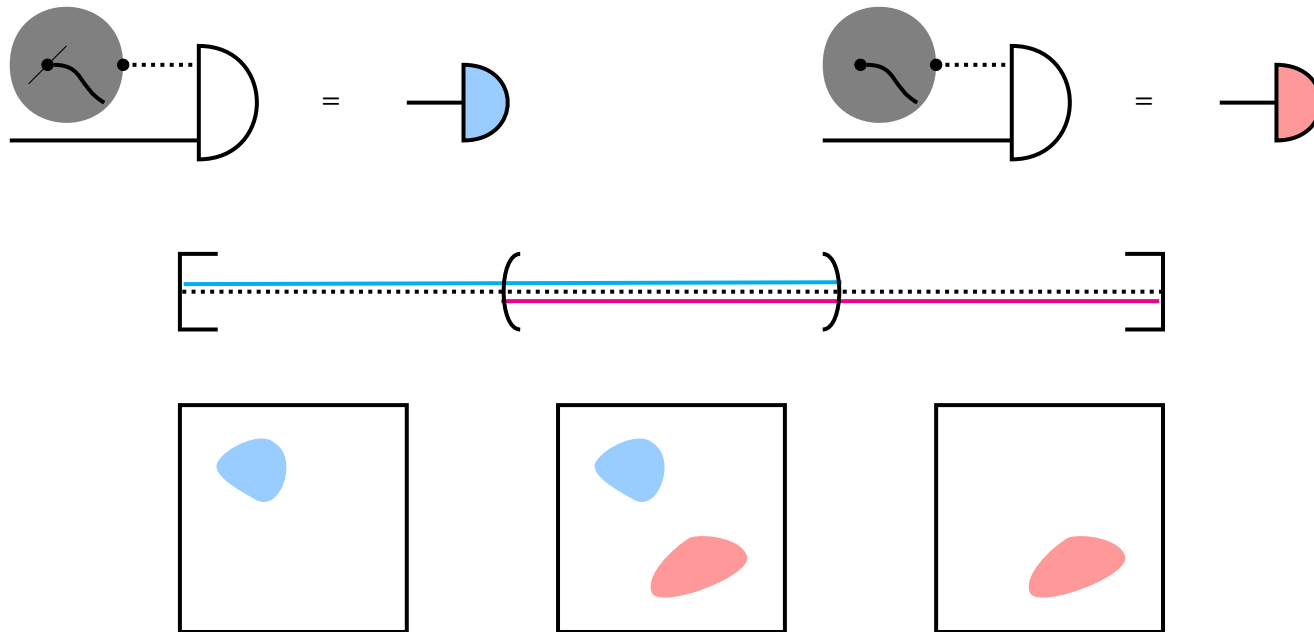
Rigid displacement

$$\begin{array}{c}
 \text{---} \bullet \text{---} = \text{---} \circ (\mathbf{x}, \theta, c)_i^- \text{---} \begin{array}{c} \cup \\ i \end{array} \begin{array}{c} \triangleleft i \\ \triangleright i \end{array} \text{---} \circ (\mathbf{x}, \theta, c)_i \text{---}
 \end{array}$$

Remark 1.1.18. Usually, when we are restricted to speaking of topological spaces and continuous functions, a homotopy is defined:

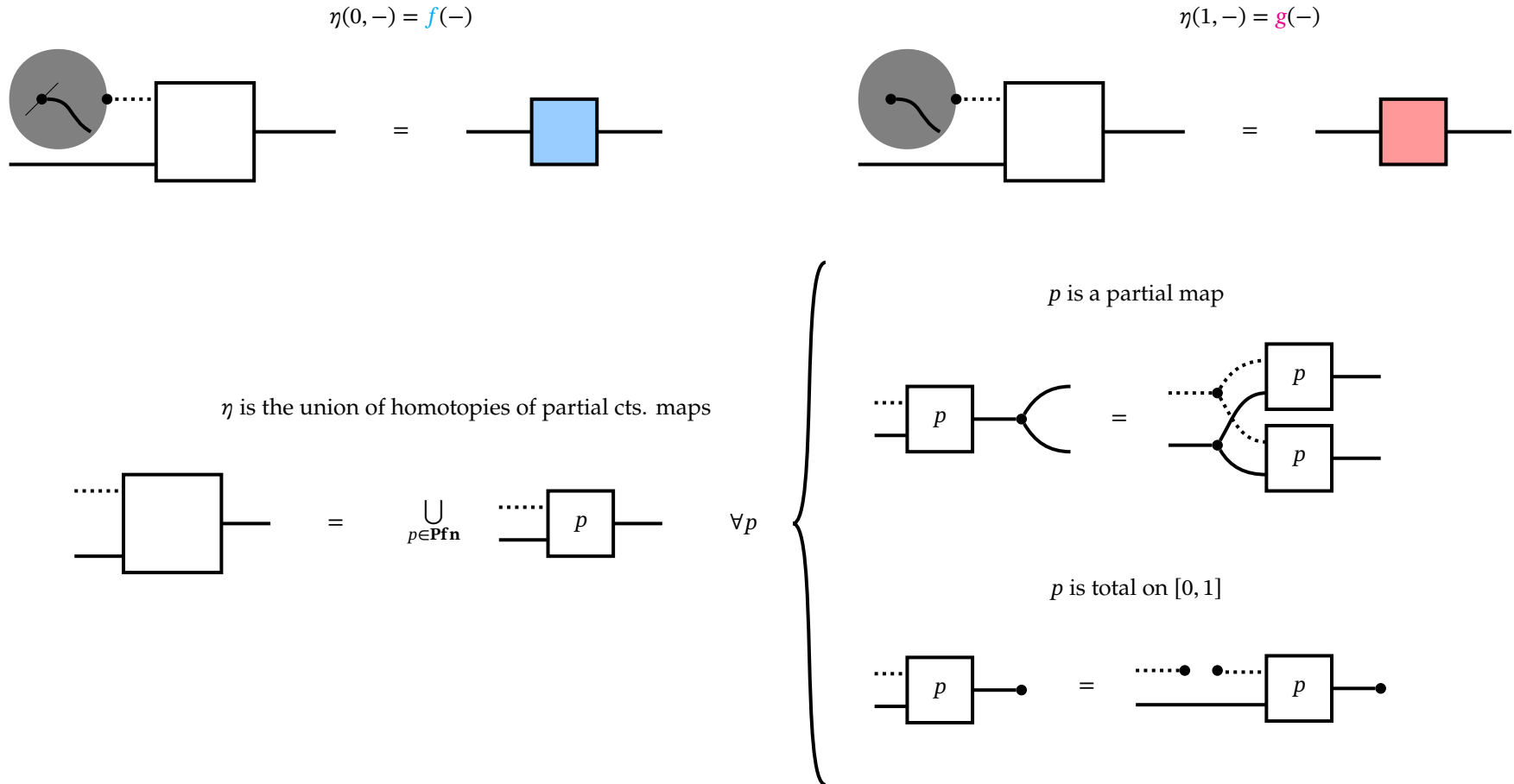
Definition 1.1.19 (Homotopy in **Top**). where f and g are continuous maps $A \rightarrow B$, a *homotopy* $\eta : f \Rightarrow g$ is a continuous function $\eta : [0, 1] \times A \rightarrow B$ such that $\eta(0, -) = f(-)$ and $\eta(1, -) = g(-)$.

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.

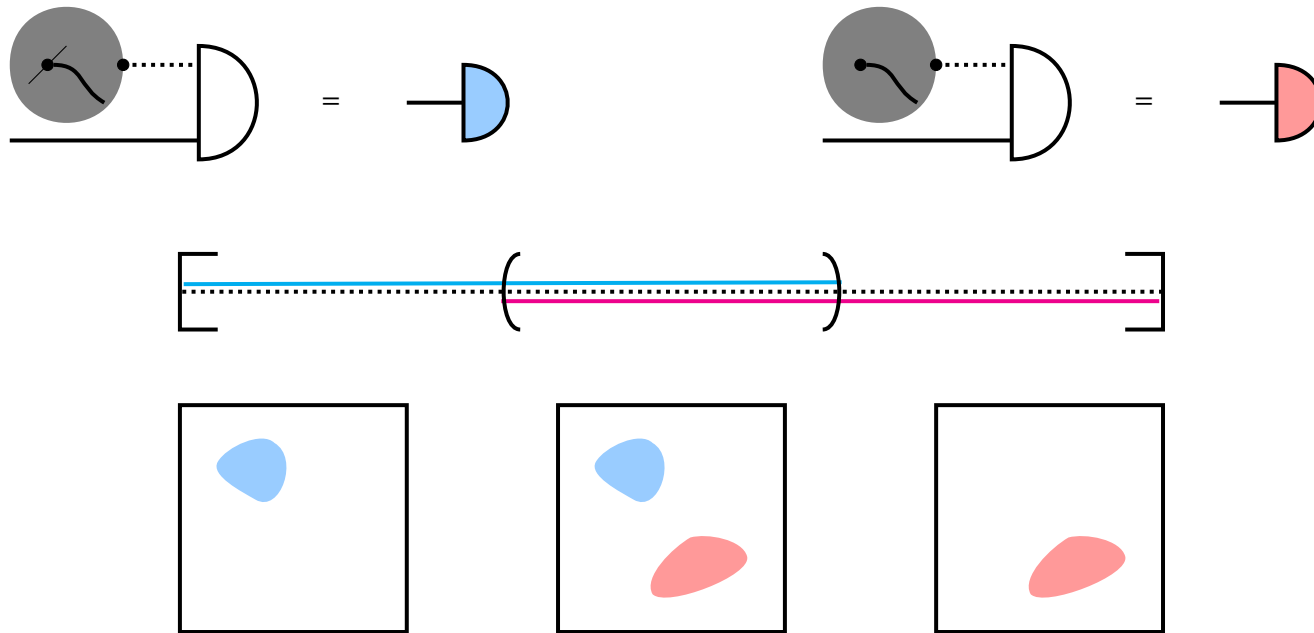


What is happening in the above film is that we have our starting open set, which stays constant for a while. Then suddenly the ending open set appears, the starting open disappears, and we are left with our ending; while *technically* there was no discontinuous jump, this isn't the notion of sliding we want. The exemplified issue is that we can patch together (by union of continuous relations) vignettes of continuous relations that are not individually total on $[0, 1]$.

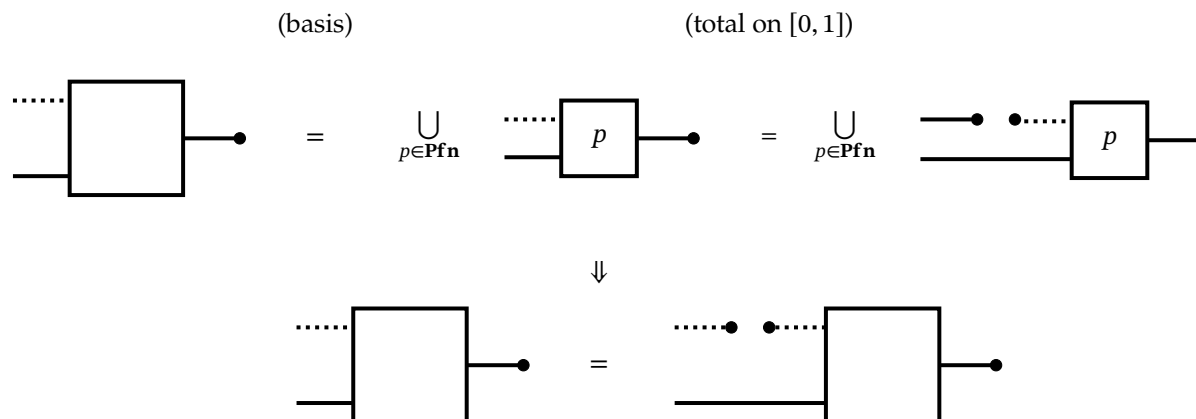
Definition 1.1.20 (Relational Homotopy). We can patch the issue in Remark 1.1.18 by asking for homotopies in **ContRel** to satisfy the additional condition that they are expressible as a union of continuous partial maps that are total on the unit interval.



Remark 1.1.21. Observe that the second condition asking for decomposition in terms of partial comes for free by Proposition ??; 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 :

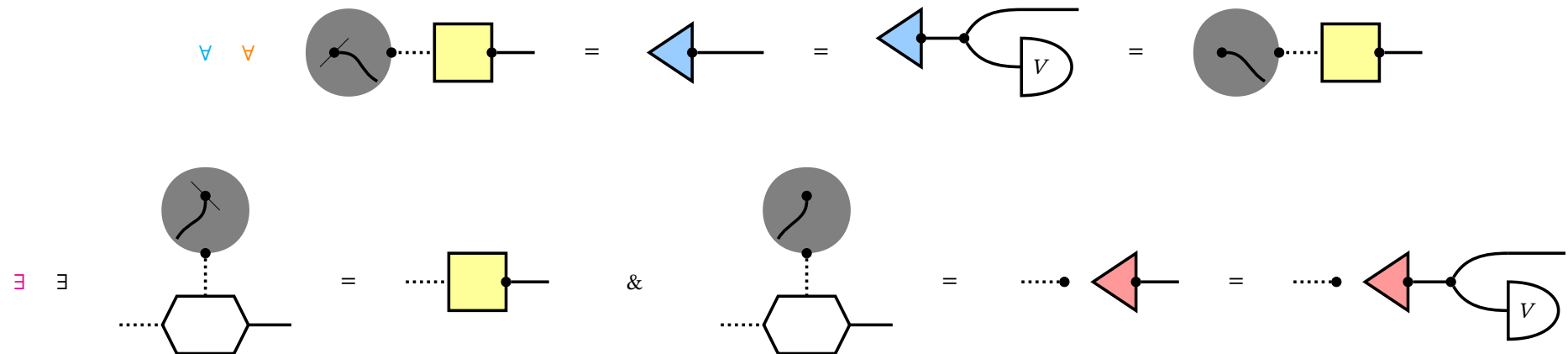


Definition 1.1.20 is "natural" in light of Proposition ??, that the partial continuous functions $A \rightarrow B$ form a basis for $\mathbf{ContRel}(A, B)$: we are just asking that homotopies between partial continuous functions – which can be viewed as regular homotopies with domain restricted to the subspace topology induced by an open set – form a basis for homotopies between continuous relations.

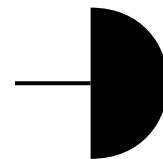


Definition 1.1.22 (Contractibility). With homotopies in hand, we can define a stronger notion of connected shapes with no holes, which are usually called *contractible*. The reason for the terminology reflects the method by which we can guarantee a shape in flatland has no holes: when any loop in the shape is *contractible* to a point.

V is *contractible* when:

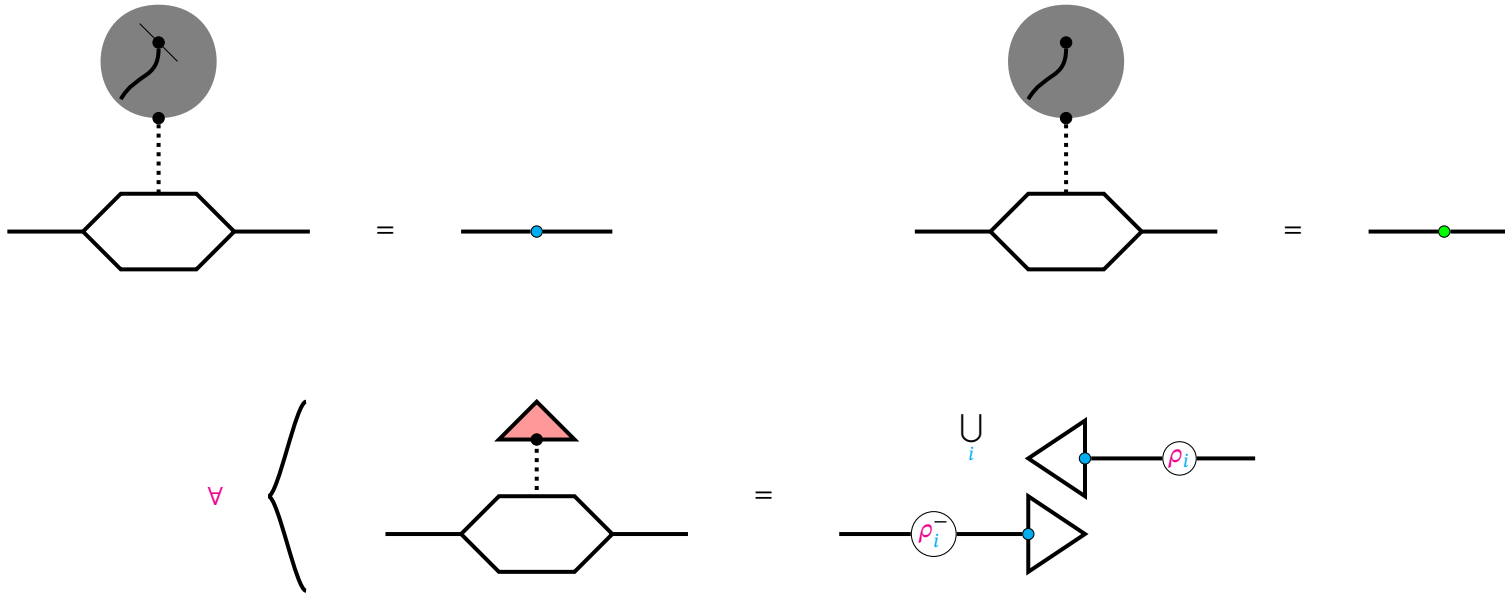


Contractible open sets are worth their own notation too; a solid black effect, this time with no hole.



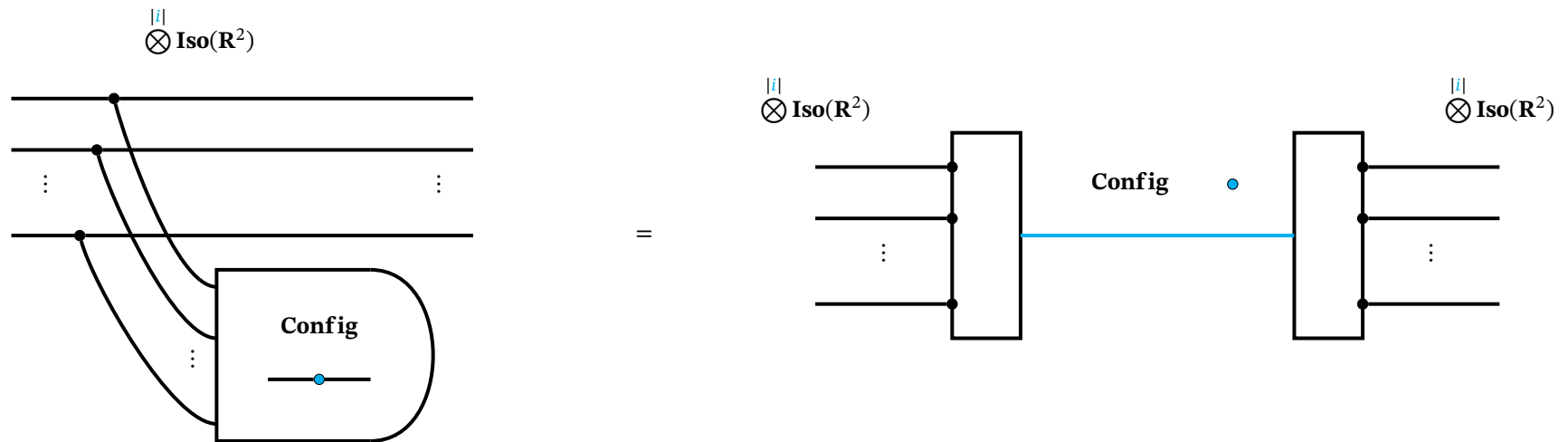
Definition 1.1.23 (Slideability in flatland). What we mean by two sticky spiders being relatable by sliding shapes is that we have a homotopy that begins at one and ends at the other, such that every point in between is itself a sticky spider related to the first by rigid displacement.

Rigid motion

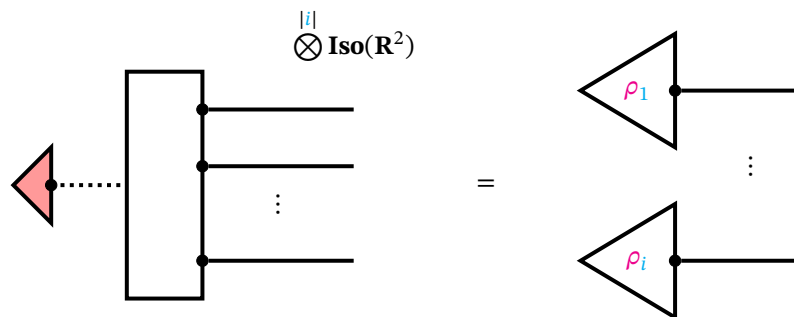


Definition 1.1.24. We can depict the *configuration space* of shapes that are obtainable by displacing the shapes of a given spider by a split idempotent through the n -fold tensor of rigid transformations – a restriction to the subspace of the largest open set contained in the subset of all valid (with correct chirality) combinations of displacements that yield another spider; points in configuration space are sticky spiders related to each other by rigid transforms. Note that as a subspace, the retract of the idempotent is a function. The section and retract pair is not unique; they may for instance encode a choice of basepoint for each shape relative to which the displacement of that shape occurs.

Configuration space of a sticky spider

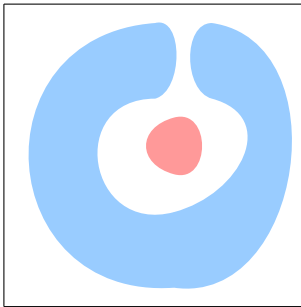


The data of rigid motion on a sticky spider as we have defined above can be captured as a continuous map from the unit interval to the space of rigid transformations; continuous paths in configuration space model motions of shapes in space.

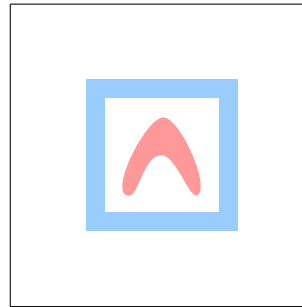


Example 1.1.25 (The connected components of configuration space). What are the connected components of configuration space? Evidently, there are pairs of spiders that are both valid displacements, but not mutually reachable by rigid motion. For example, shapes might *enclose* or *trap* other shapes, or shapes might be *interlocked*. Depicted are some pairs of configurations that are mutually unreachable by rigid transformations. Now we have the conceptual toolkit to begin modelling these concepts in the configuration space of a sticky spider.

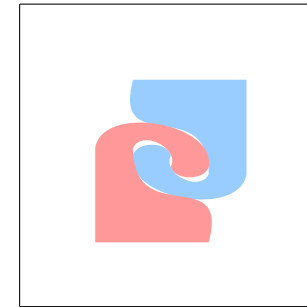
Trapped



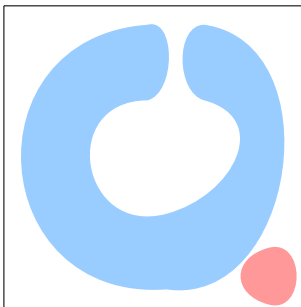
Enclosed



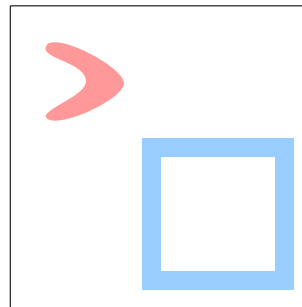
Interlocked



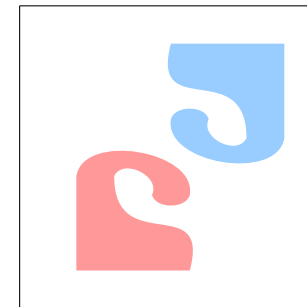
Not trapped



Not enclosed

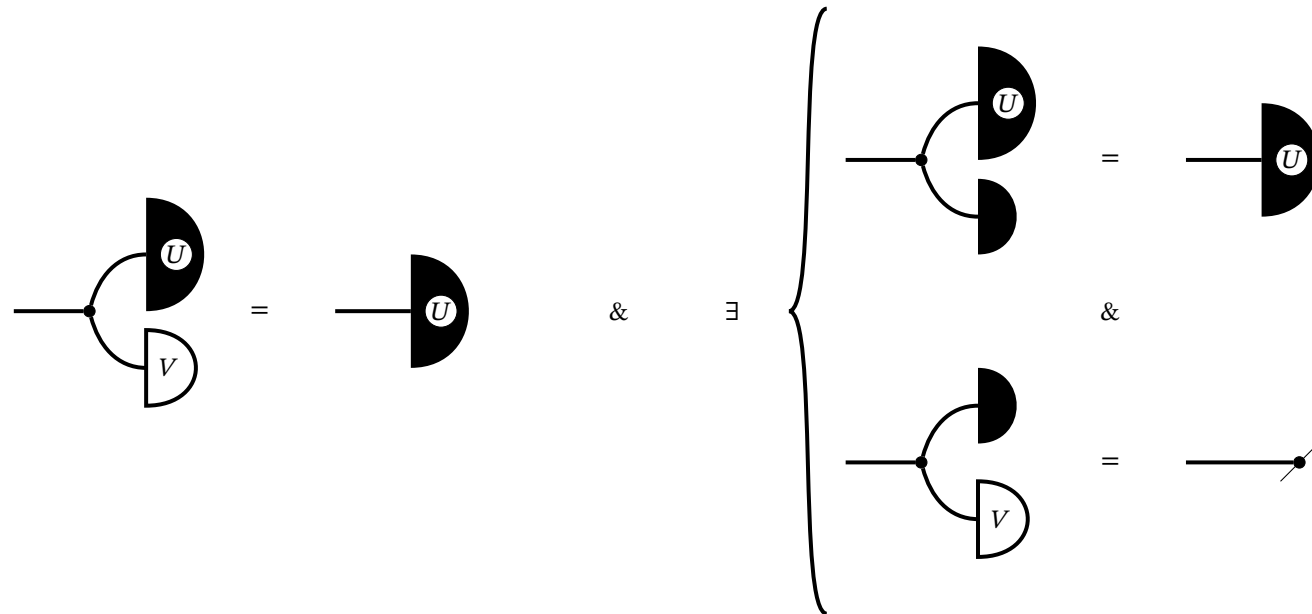


Not interlocked



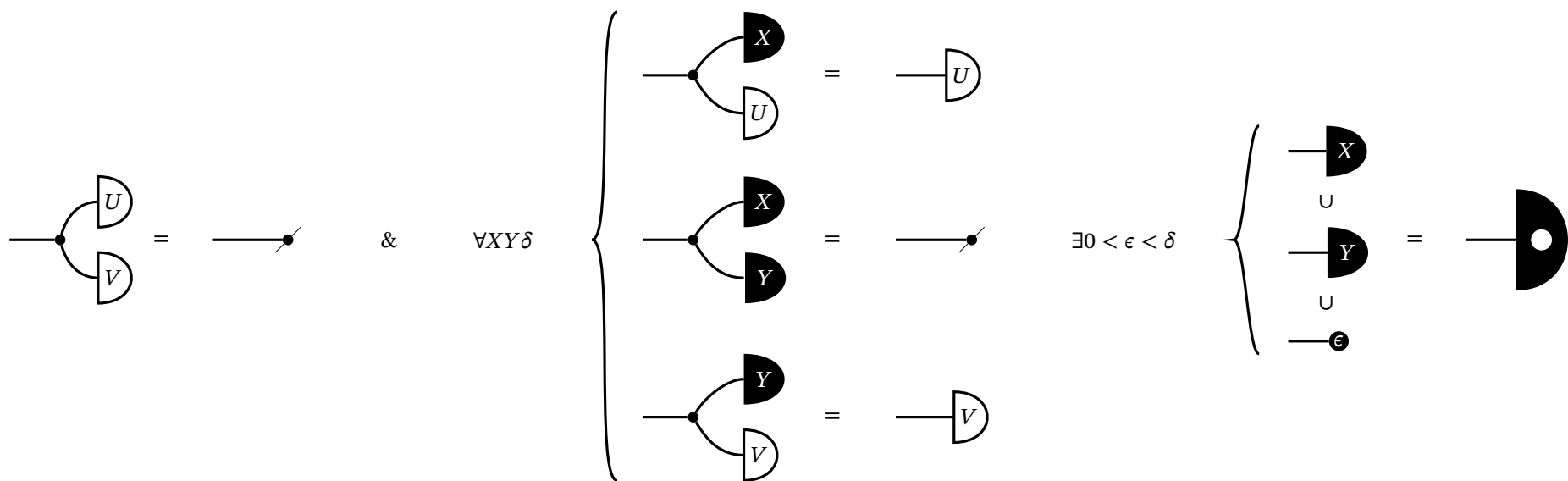
Definition 1.1.26 (Parthood). Let's say that a "part" refers to an entire simply connected component. Simply connected is already a concept in our toolkit. A shape U is disjoint from another shape V intuitively when we can cover U in a blob with no holes such that the blob has no overlap with V . So, U is a part of V when it is simply connect, wholly contained in V , and there exists a contractible open that is disjoint from V that covers U . Diagrammatically, this is:

U is a part of V



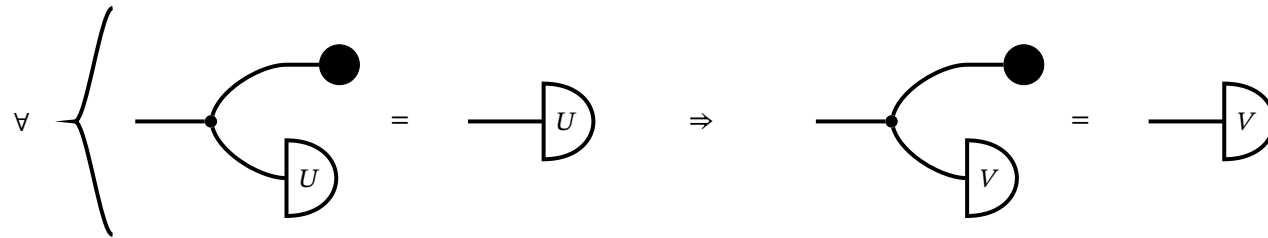
Definition 1.1.27 (Touching). Let's distinguish touching from overlap. Two shapes are "touching" intuitively when they are as close as they can be to each other, somewhere; any closer and they would overlap. Let's assume that we can restrict our attention to the parts of the shape that are touching, and that we can fill in the holes of these parts. At the point of touching, there is an infinitesimal gap – just as when we touch things in meatspace, there is a very small gap between us and the object due to the repulsive electromagnetic force between atoms. To deal with infinitesimals we borrow the $\epsilon - \delta$ trick from mathematical analysis; for any arbitrarily small δ , we can pick an even smaller ball of radius ϵ such that if we stick the ball in the gap, the ball forms a bridge that overlaps the two filled-in shapes, which allows us to draw a continuous line between them. Diagrammatically, this is:

U and V are *touching*



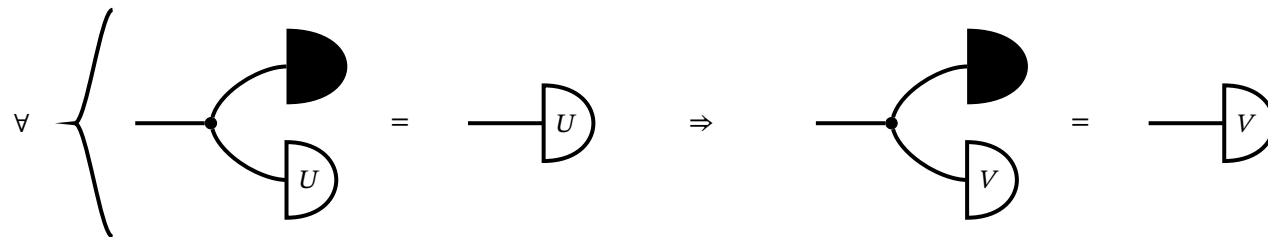
Definition 1.1.28 (Within and surrounds). If U surrounds V , or equivalently, if V is within U , then we are saying that leaving V in almost any direction, we will see some of U before we go off to infinity. We can once again use open balls for this purpose, which correspond to possible places you can get to from a starting point \mathbf{x} within a distance ϵ . In prose, we are asking that any open ball that contains all of U must also contain all of V .

V is within U , or U surrounds V



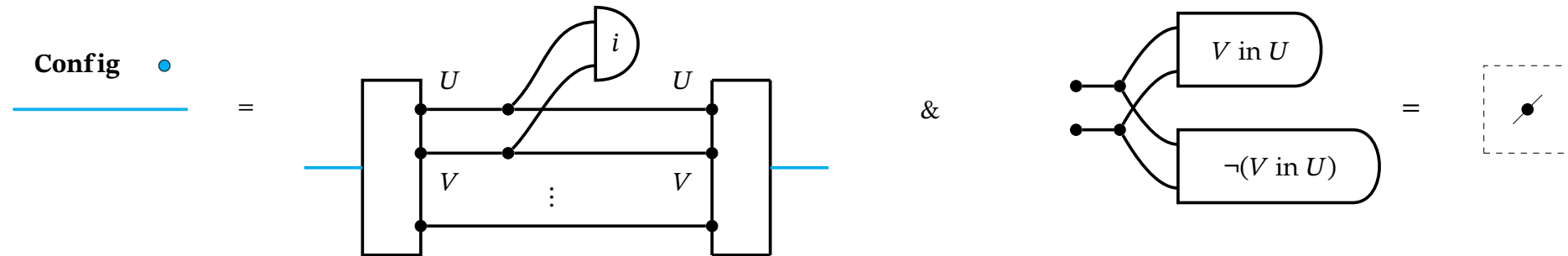
Definition 1.1.29 (Containment and enclosure). There is a strong version of within-ness, which we will call enclosure. As in when we are in elevators and the door is shut, nothing gets in or out of the container. Intuitively, there is a hole in the container surrounded on all sides, and the contained shape lives within the hole. To give a real-world example, honey lives within a honeycomb cell in a beehive, but whether the honey is enclosed in the cell depends on whether it is sealed off from air with beeswax. So in prose we are asking that any way we fill in the holes of the container with a blob, that blob must cover the contained shape. Diagrammatically, this amounts to levelling up from open balls in our previous definition to contractible sets:

U encloses V



Definition 1.1.30 (Trapping). There is an intermediate notion between within-ness and enclosure; for instance, standing in the stonehenge you are surrounded by the pillars, but you can always walk away, whereas if the pillars are very close, such as the bars of a jail cell, a human would not be able to leave the trap while still being able to see the outside. The difficulty here is that relative sizes come into play: small animals would still consider it a case of mere within-ness, because they can still walk away between the bars. So we would like to say that no matter how the pair of objects move rigidly, being trapped means that the trapped V stays within U . In other words, that in configuration space, if we forget about all other shapes, we can partition our space of configurations by two concepts, whether V is within U or not, and moreover that these two components is disjoint – i.e. not simply connected – so there is no rigid motion that can allow V to escape from being within U if V starts off trapped inside in U .

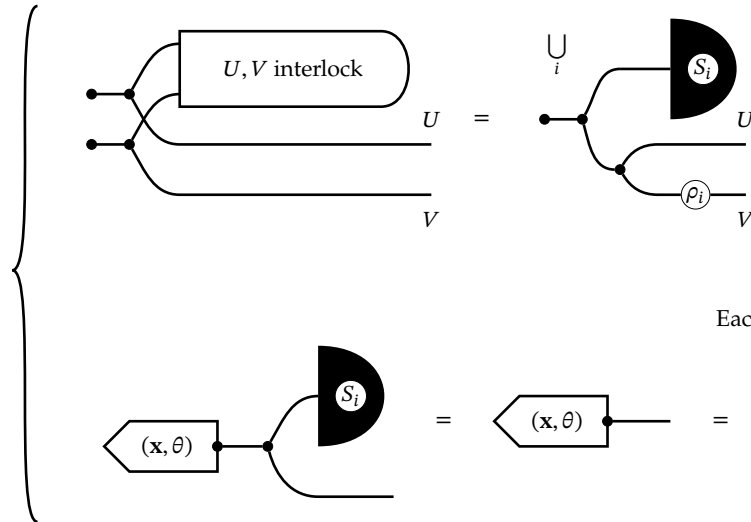
$$\bigcup_{i \in \{V \text{ in } U, \neg(V \text{ in } U)\}}$$



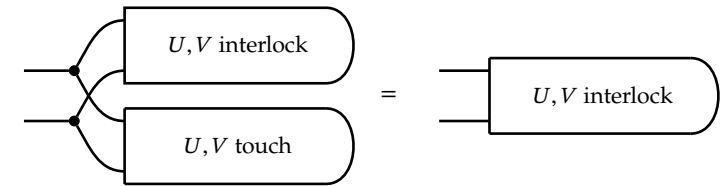
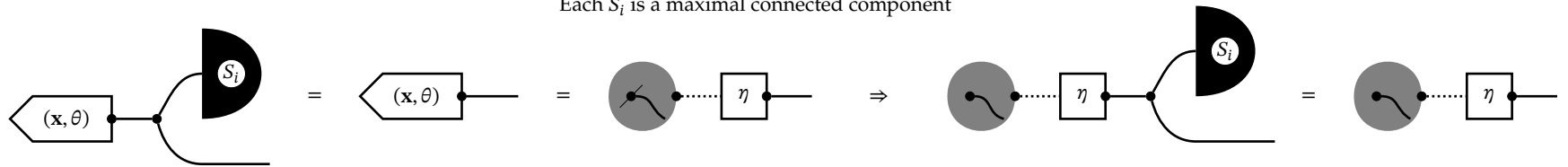
Definition 1.1.31 (Interlock). Two shapes might be tightly interlocked without being inside one another. Some potentially familiar examples are plastic models of molecular structure that we encounter in school, metal lids in cold weather that are too tightly hugging the glass jar, or stubborn Lego pieces that refuse to come apart. The commonality of all these cases is that the two shapes must move together as one, unless deformed or broken. In other words, when two shapes are interlocked, knowing the position in space of one shape determines the position of the other, and this determination is a fixed isometry of space. So we only need to specify a range of positions S for the entire subconfiguration of interlocked shapes U and V , and we may obtain their respective positions by a fixed rigid motion ρ . Since objects may interlock in multiple ways, we may have a sum of these expressions. We additionally observe that interlocking shapes should also be touching, which translates to containment inside the touching concept. Finally, we observe that as in the case of entrapment and enclosure, rigid motions are interlocking-invariant, which translates diagrammatically to the constraint that each S, ρ expression is an entire connected component in configuration space.

 $\exists S_i \rho_i \forall (\mathbf{x}, \theta) \eta$

Determined positions

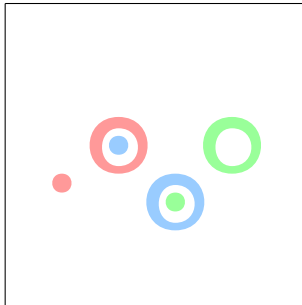


Interlock entails touching

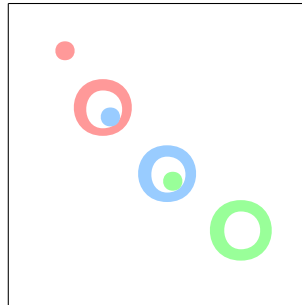

Each S_i is a maximal connected component


Definition 1.1.32 (Mutual constraint). A weaker notion of interlocking is when shapes only imperfectly determine each other's potential displacements, by specifying an allowed range. It would be an understatement to say there is some interest in studying how shapes mutually constrain each other's movements in this way.

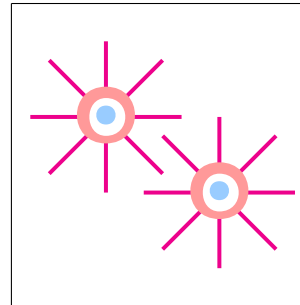
Chain (loose)



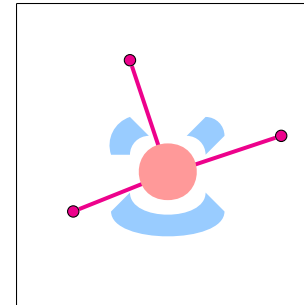
Chain (taut)



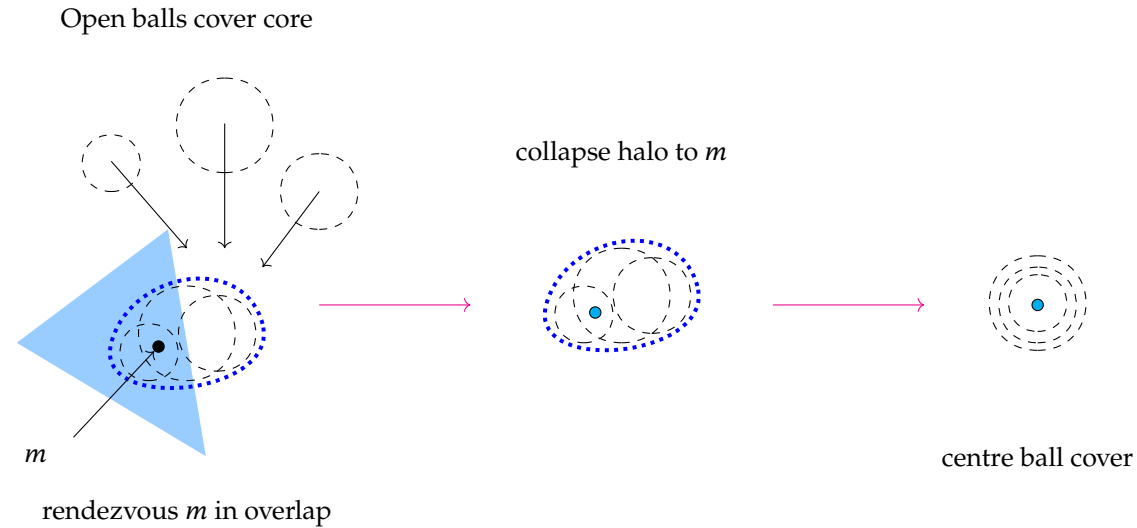
Gears



Switch



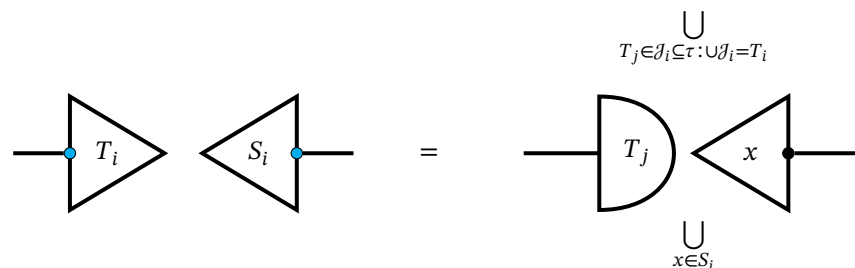
Construction 1.1.33 (Morphing sticky spiders with homotopies). We aim to construct homotopies relating (almost) arbitrary sticky spiders. For now we focus on just changing one shape into another arbitrary one. The idea is as follows. First, we need a cover of open balls $\cup \mathcal{J} = T^0$ and $\cup \mathcal{K} = T^1$ of the start and end cores T^0 and T^1 such that each $k \in T^1$ is expressible as a rigid isometry of some core $j \in \mathcal{J}$; this is so we can slide and rearrange open balls comprising T^0 and reconstruct them as T^1 . As an intermediate step to eliminate holes and unify connected components, we gather all of the balls at a meeting point m (to be determined shortly.) Intuitively we can illustrate this process as follows:



Second, in order to perform the sliding of open balls, we observe that, given a basepoint to act as origin (which we assume is provided by the data of the split idempotent of configuration space) we can express the group action of rigid isometries $\mathbf{Iso}(\mathbb{R}^2)$ on \mathbb{R}^2 as a continuous function:

$$\begin{array}{c}
 \mathbf{Iso}(\mathbb{R}^2) \\
 \hline
 \hline
 \mathbb{R}^2
 \end{array}
 \xrightarrow{\quad \mathbb{R}^2 \quad}
 ((\mathbf{a}, \theta), \mathbf{b}) \mapsto \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mathbf{b} + \mathbf{a}$$

Third, before we begin sliding the open balls, we must ensure that the halo of the shape cooperates. We observe that a given shape i in a sticky spider may be expressed as the union of a family of constant continuous partial functions in the following way. Given an open cover \mathcal{J} such that $\bigcup \mathcal{J} = T_i$, where T_i is the core of the shape i , each function is a constant map from some $T_j \in \mathcal{J}$ to some point $x \in S_i$, where S_i is the halo of the shape i . For each $T_j \in \mathcal{J}$ and every point $x \in S_i$, the constant partial function that maps T_j to x is in the family.



By definition of sticky spiders, there must exist some point m that is in both the core and the halo: we pick such a point as the rendezvous for the open balls. For each partial map in the family, we provide a homotopy that varies only the image point x continuously in the space to finish at m . Now we can slide the open balls to the rendezvous m . Since homotopies are reversible by the continuous map $t \mapsto (1 - t)$ on the interval, we can perform the above steps for shapes T^0 and T^1 to finish at the same open ball, reversing the process for T^1 and composing sequentially to obtain a finished transformation. The final wrinkle to address is when dealing with multiple shapes. Recalling our exclusion conditions (Propositions ??) for shapes, it may be that parts of one shape are enclosed in another, so the processes must be coordinated so that there are no overlaps. For example, the enclosing shape must be first opened, so that the enclosed shape may leave. I will keep it an article of faith that such coordinations exist. I struggle to come up with a proof that all sticky spiders \mathbf{R}^2 are mutually transformable by homotopy in this (or any other) way, so that will remain a conjecture. But it is clear that a great deal of spiders are mutually transformable; almost certainly any we would care to draw. So this will just be a construction for now.

1.2 Interpreting text circuits in *ContRel*

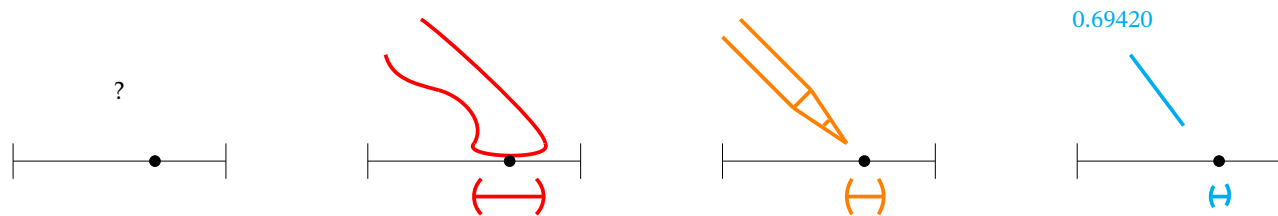
Recall that there were some mathematically odd choices in conventions for text circuits as syntactic objects, such as the labelling of noun wires and some subtleties with respect to interchange of parallel gates and text order. The explanation of those choices was deferred "to semantics", and since we're 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*.

1.2.1 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 are 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.

1.2.2 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 1.1 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.



This may explain 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

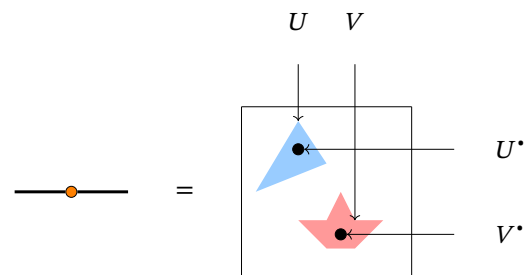
Figure 1.1: Points in space are a useful mathematical fiction. Suppose we have a point on a unit interval. Consider how we might tell someone else about where this point is. We could point at it with a pudgy appendage, or the tip of a pencil, or give some finite decimal approximation. But in each case 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

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 permissible 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.

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.

1.2.3 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 representations¹. 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 1.1.24) 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.

1.2.4 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.

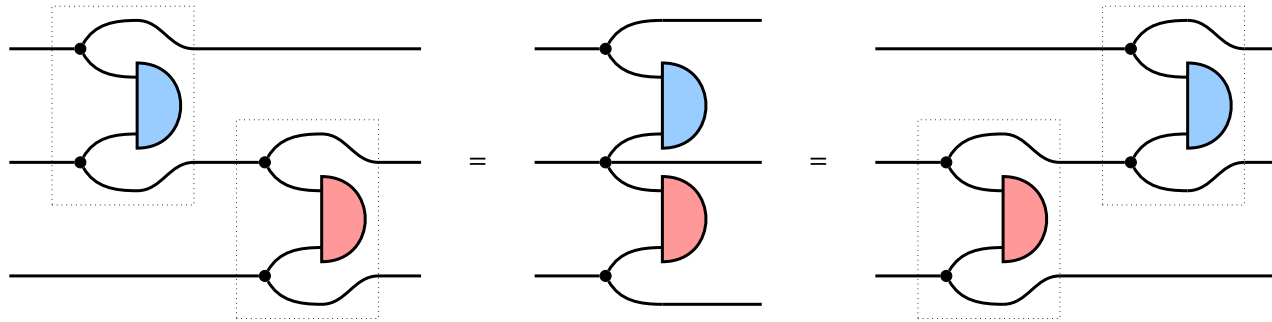


Figure 1.2:

Example 1.2.2 (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.

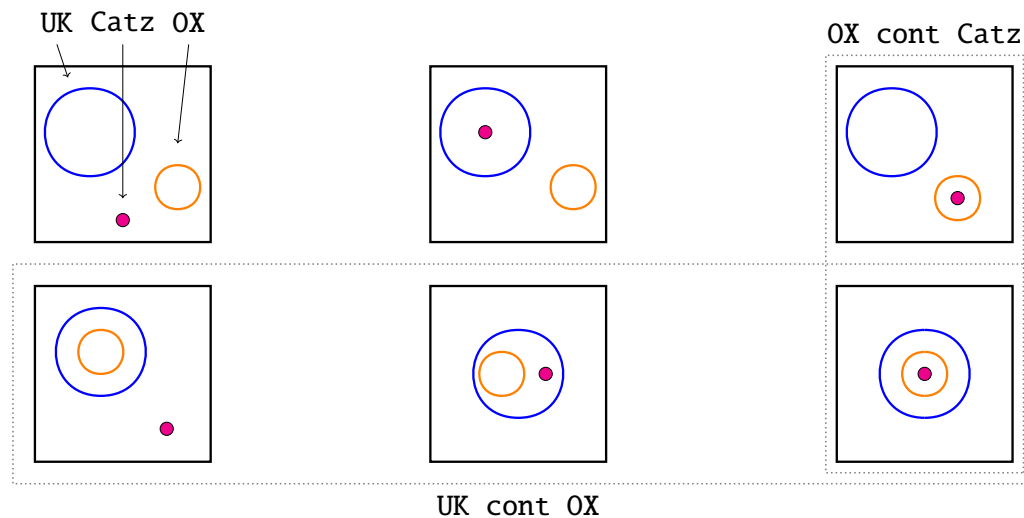


Figure 1.3: Consider the configuration space of a sticky spider on the unit square with three labelled shapes, which has 6 connected components, depicted. Oxford contains Catz. restricts away configurations where Catz is not enclosed in Oxford. Adding on England contains Oxford. further restricts away 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.

1.2.5 Homotopies: dynamic verbs and weak interchange

Figure 1.4: *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 1.2.4 (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 1.5: 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 may define the interior $i(\nabla)$ 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.

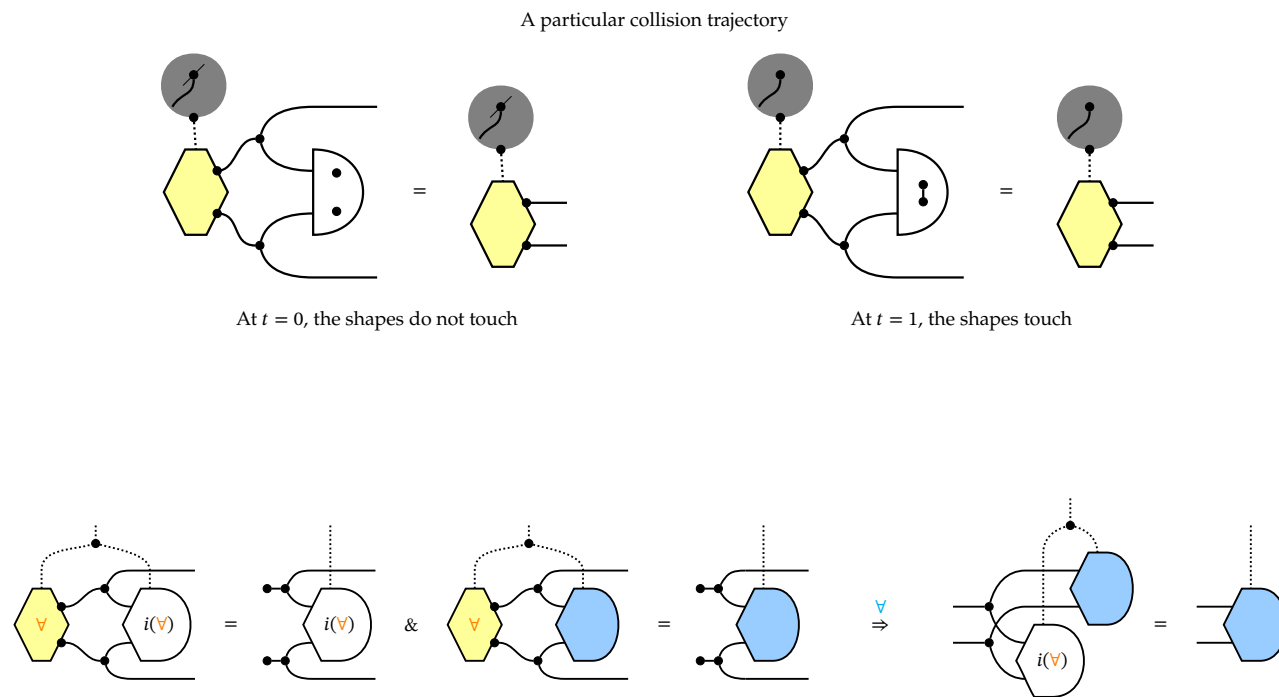
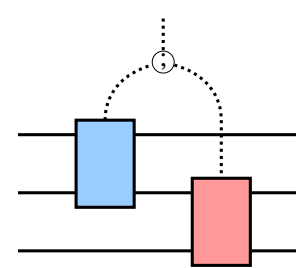
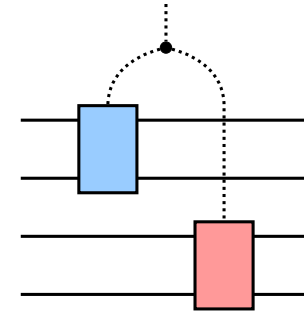


Figure 1.6: Collision blocks the commutation of the statives touching and not touching. In prose, the text Alice and Bob are not touching. Alice and Bob collide. Alice and Bob are touching. is not equal to Alice and Bob are (touching and not touching). Alice and Bob collide.; the latter is an incongruent state of affairs, which is reflected by an empty set of potential models (when touching and not touching intersect.)

Figure 1.7: 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.

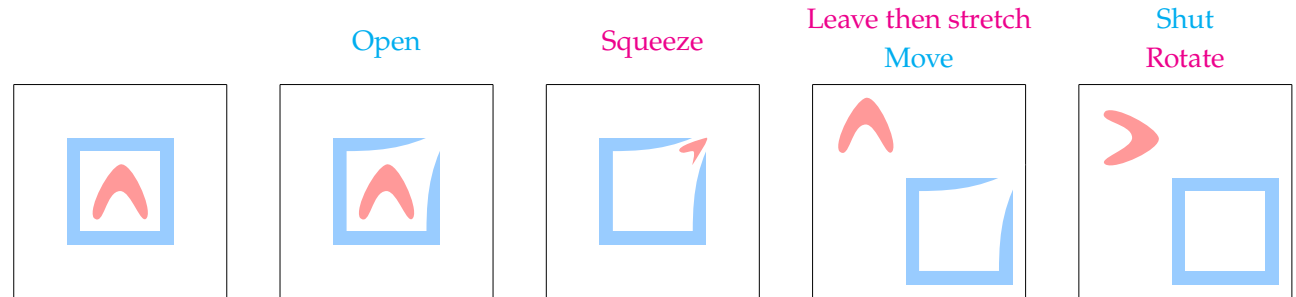


Sequential composition of motions



Parallel composition of motions

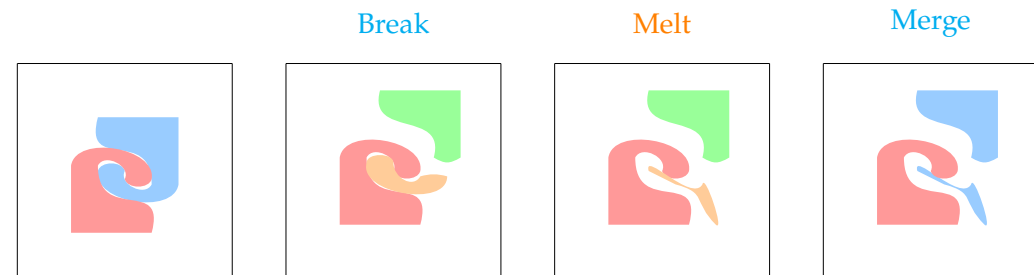
Figure 1.8: What kinds of actions are there? In our toy setting, in general we can define actions that arbitrarily change states of affairs if we do not restrict ourselves to rigid motions. The trick to doing this is the observation that arbitrary homotopies allow deformations, so our verb gates allow shapes to shrink and open and bend in the process of a homotopy, as long as at the end they arrive at a rigid displacement of their original form.



Enclosed

Not enclosed

Figure 1.9: We can further generalise by noting that completely different spiders can be related by homotopy, so we can model a situation where there is a permanent bend, or how a rigid shape might shatter. Construction 1.1.33 shows that dynamic verbs can be constructed to respect a broad range of pre- and postconditions expressed in terms of statives.



Interlocked

Not interlocked

1.2.6 Coclosure: adverbs and adpositions

Figure 1.10: 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.

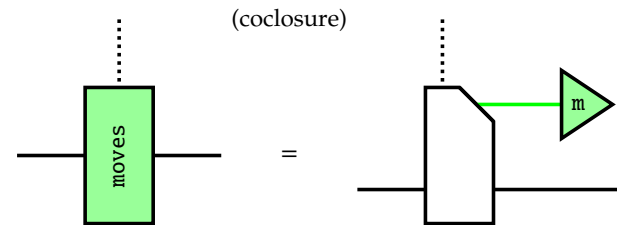


Figure 1.11: Adverb-boxes may be modelled as static restrictions in movement-space. For instance, *straight* may restrict movements to just those that satisfy some notion of path-length minimality: e.g., given a metric in movement-space on path-lengths, we may construct an open ball (Definition 1.1.14) around the geodesic to model the adverb *straight*.

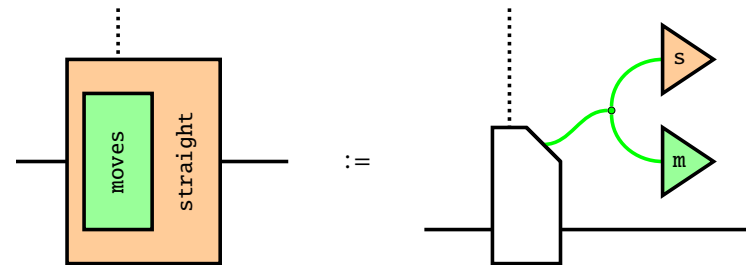
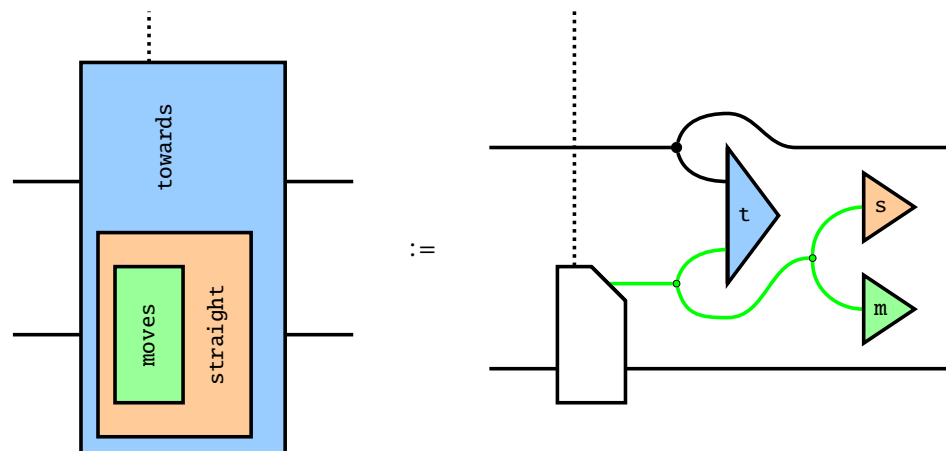


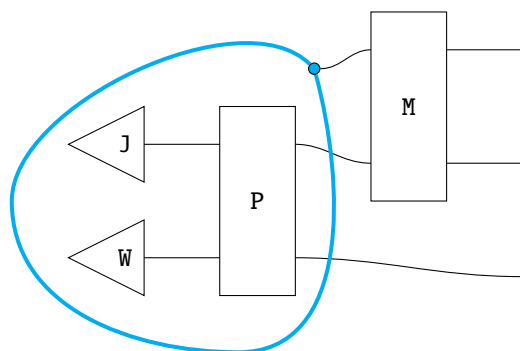
Figure 1.12: Similarly, adposition-boxes may be modelled as static restrictions on the product of the spaces of nouns and verbs. For instance, *towards* may be modelled as an open set that pairs potential positions of the thing-being-moved-towards with movements in movement-space that indeed move towards the target.



1.2.7 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.



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

Figure 1.13:

Example 1.2.6.

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.

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 1.14:

Definition 1.2.8 (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 $ev_B^A : A \otimes \Xi \rightarrow B$, which takes code-format (which we will just denote Ξ going forward) as an additional input, and for every possible program $f : A \rightarrow B$, there exists a code-state $\ulcorner f \urcorner : I \rightarrow \Xi$ that evaluates to the desired program.

$$\forall A, B \in Ob(\mathcal{C}) \exists ev_B^A : A \otimes \Xi \rightarrow B \forall f : A \rightarrow B \exists \ulcorner f \urcorner : I \rightarrow \Xi$$

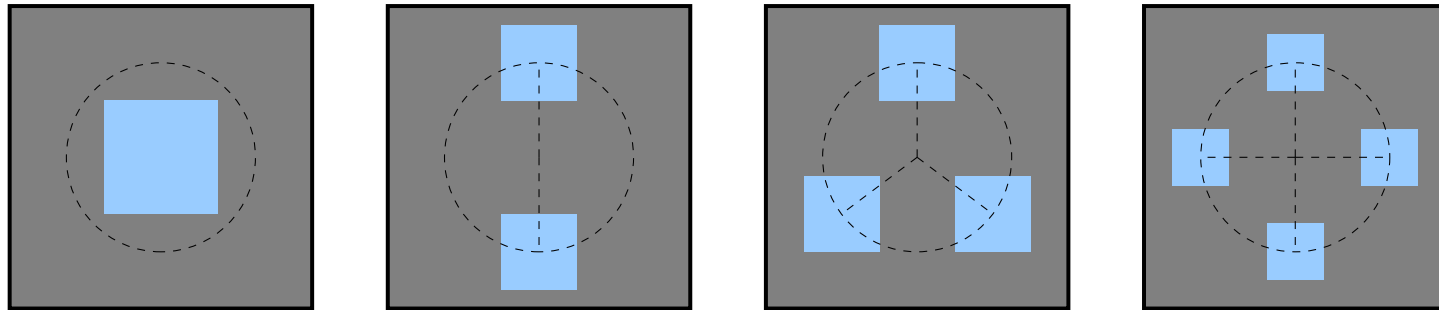
It would be true but unhelpful to conclude that any programming language is a model for text circuits, using the code data format as the noun wire. In **ContRel**, the unit square suffices.

Scholium 1.2.9. 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 $A \multimap B$ or B^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 Pavlovic’s formalism because I prefer string diagrams to commuting diagrams.

Construction 1.2.10 (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 1.2.16.)

Proposition 1.2.11 ($((0, 1) \times (0, 1)$ splits through any countable set X). For any countable set X , the open unit square \mathbb{I} has a sticky spider that splits through X^* .

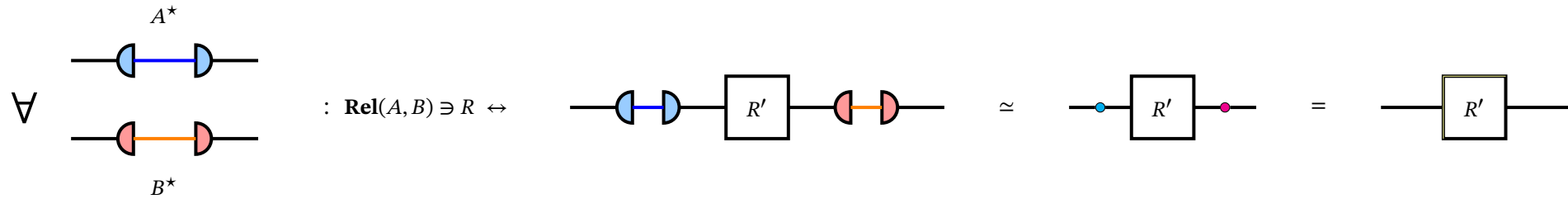
Proof. The proof is by construction. We'll assume the sticky-spiders to be mereologies, so that cores and halos agree. Then we only have to highlight the copiable open sets. Take some circle and place axis-aligned open squares evenly along them, one for each element of X . The centres of the open squares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



□

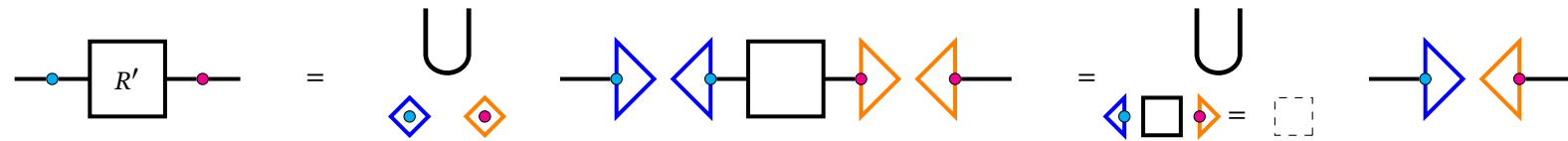
Definition 1.2.12 (Morphism of sticky spiders). A morphism between sticky spiders is any morphism that satisfies the following equation.

Proposition 1.2.13 (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 $R : A \rightarrow B$.



Proof.

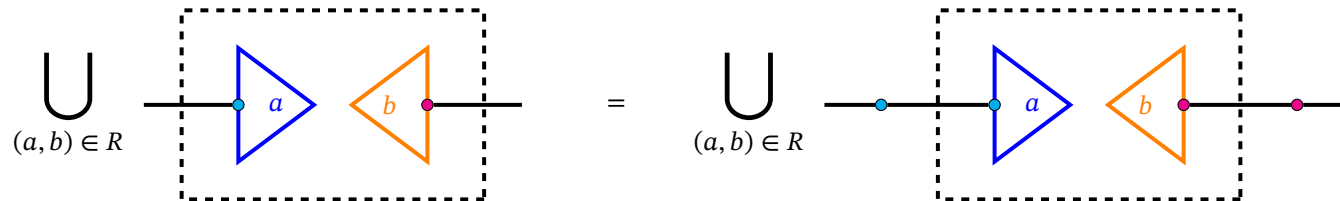
(\Leftarrow) : Every morphism of sticky spiders corresponds to a relation between sets.



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

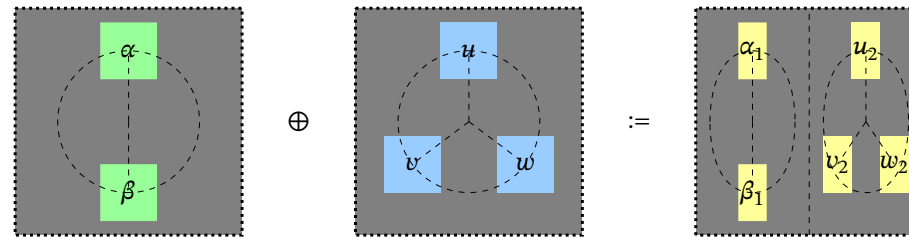


(\Rightarrow) : By idempotence of (co)copiabes, every relation $R \subseteq A \times B$ corresponds to a morphism of sticky spiders.



□

Construction 1.2.14 (Representing sets in their various guises within \boxtimes). We can represent the direct sum of two \boxtimes -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.

$$\begin{array}{cccc}
 \text{---} \text{---} \text{---} & \text{---} \text{---} \text{---} & \text{---} \text{---} \text{---} & \text{---} \text{---} \text{---} \\
 (x, y) \mapsto \left(\frac{x}{2}, y\right) & (x, y) \mapsto \left(\frac{x+1}{2}, y\right) & (x, y)|_{x < \frac{1}{2}} \mapsto (2x, y) & (x, y)|_{x > \frac{1}{2}} \mapsto (2x - 1, y)
 \end{array}$$

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

$$\text{---} \text{---} \text{---} = \text{---} = \text{---} \text{---} \text{---}$$

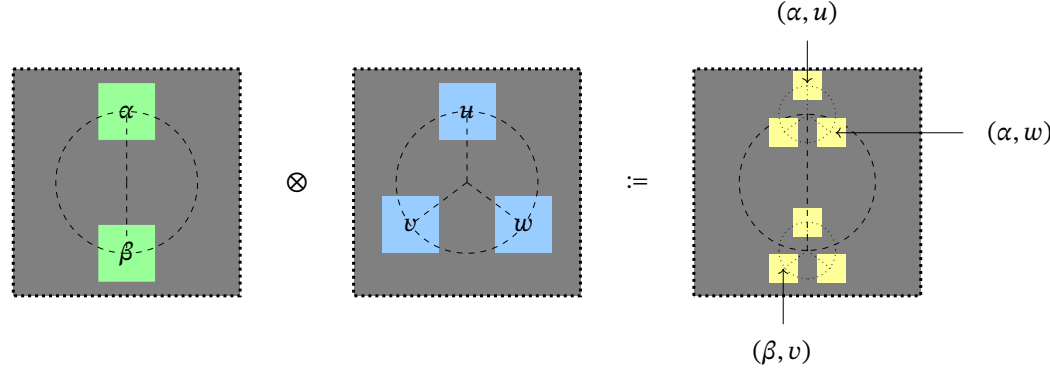
Now, to put the two halves together and to take them apart, we introduce the following two relations. In tandem with the squishing and stretching we have defined, these will behave just as the projections and injections for the direct-sum biproduct in **Rel**.

$$\begin{array}{ccc}
 \text{---} \text{---} \text{---} & := & \text{---} \text{---} \text{---} \cup \text{---} \text{---} \text{---} \\
 \text{---} \text{---} \text{---} & := & \text{---} \text{---} \text{---} \cup \text{---} \text{---} \text{---}
 \end{array}$$

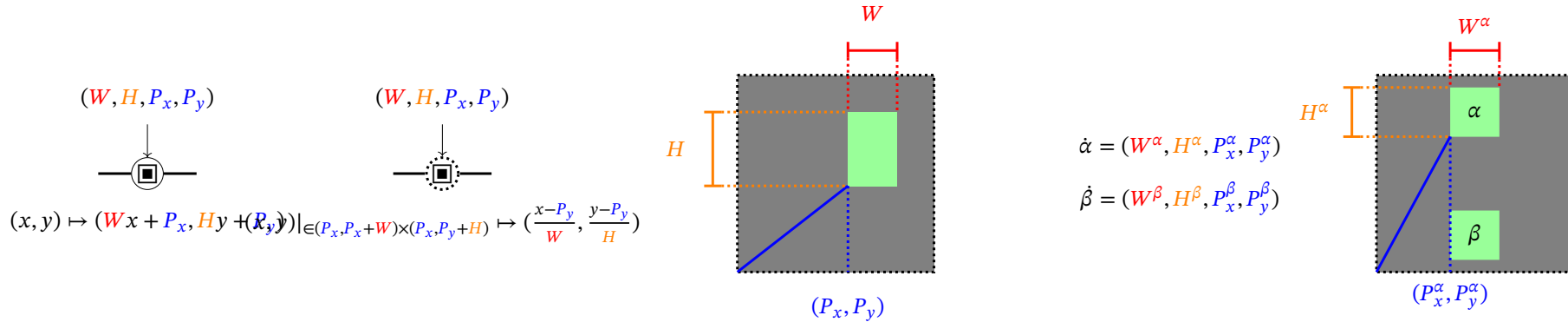
The following equation tells us that we can take any two representations in \boxtimes , put them into a single copy of \boxtimes , and take them out again.

$$\text{---} \text{---} \text{---} = \text{---} \text{---} \text{---}$$

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

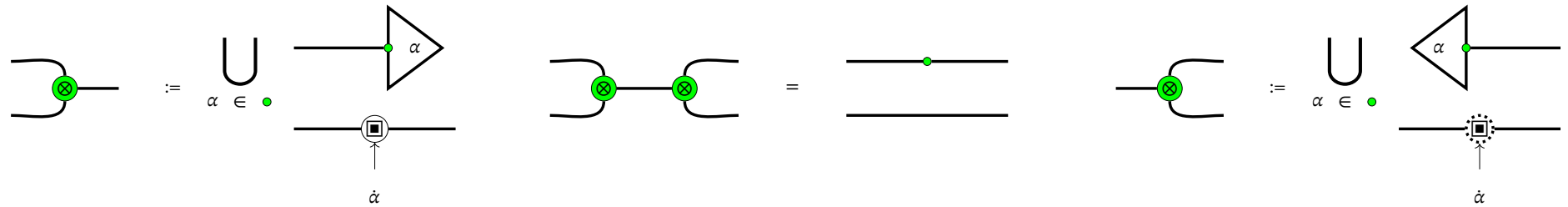


The important bit of technology here is a family of homeomorphisms of \mathbb{R}^2 parameterised by axis-aligned open boxes, that allow us to squish and stretch spaces. Thus for every representation of a set in \mathbb{R}^2 by a sticky-spider, where each element corresponds to an axis-aligned open box, 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.

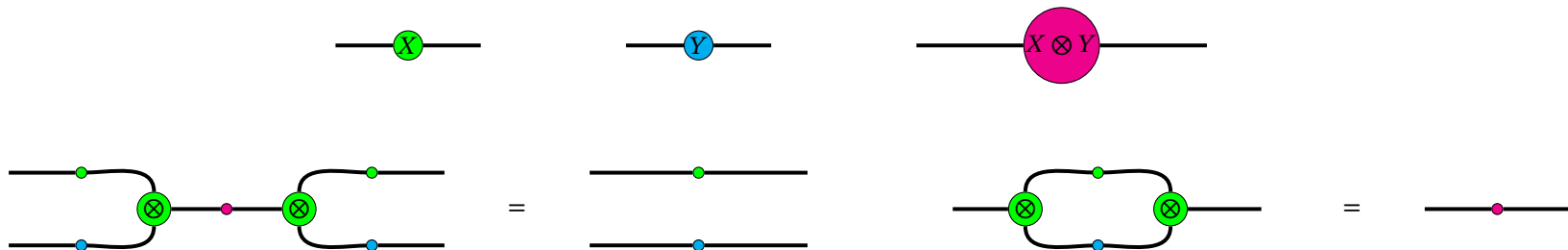


Remark 1.2.15. The essential idea is that the whole of the unit square is homeomorphic to part of it. In particular this means (modulo a point), we may make copies of shapes outside a container that are in homeomorphic correspondence with shapes within a container, the classic example being thought bubbles 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 1.2.16 operate directly on set-theoretic representations of mental contents.

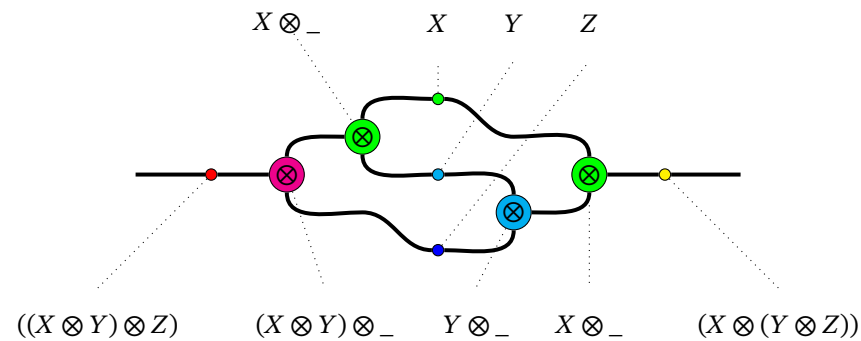
Now we can define the "tensor X on the left" relation $_ \rightarrow X \otimes _$ and its corresponding cotensor.



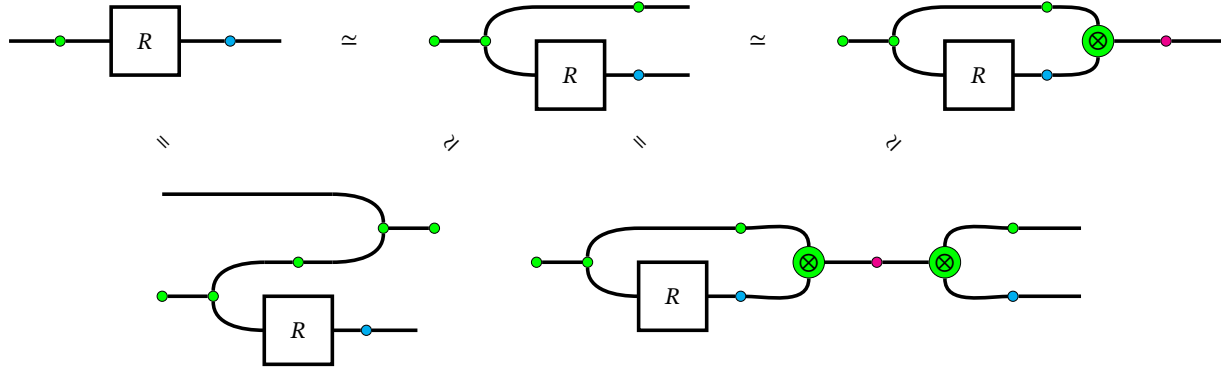
The tensor and cotensor behave as we expect from proof nets for monoidal categories.



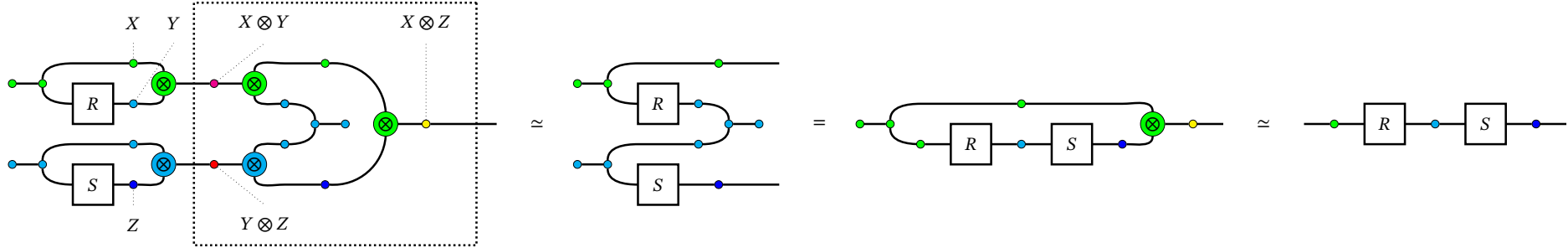
And by construction, the (co)tensors and (co)pluses interact as we expect, and they come with all the natural isomorphisms between representations we expect. For example, below we exhibit an explicit associator natural isomorphism between representations.



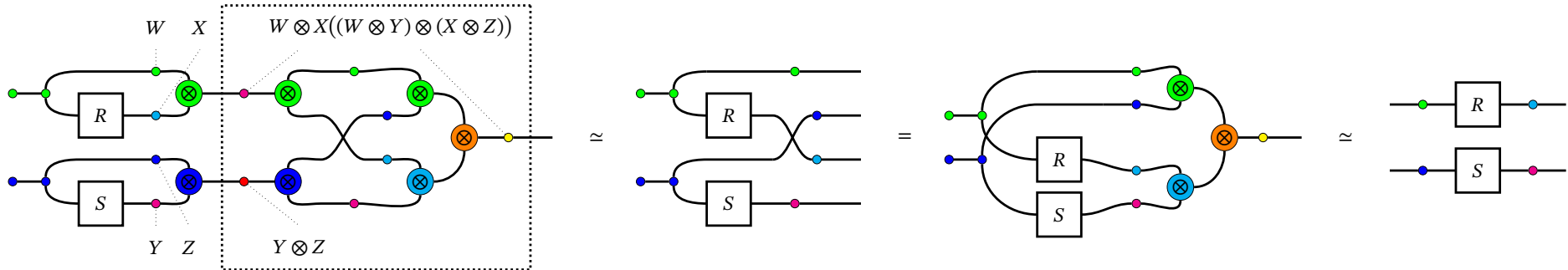
Construction 1.2.16 (Representing relations between sets and their composition within \mathbb{M}). With all the above, we can establish a special kind of process-state duality; relations as processes are isomorphic to states of \mathbb{M} , up to the representation scheme we have chosen. This is part 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 sequential 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].
- [Fri05] Harvey Friedman. [FOM] Characterization of R/Simple proof, February 2005.
- [HS20] Nathan Haydon and Paweł Sobociński. Compositional Diagrammatic First-Order Logic. In Ahti-Veikko Pietarinen, Peter Chapman, Leonie Bosveld-de Smet, Valeria Giardino, James Corter, and Sven Linker, editors, *Diagrammatic Representation and Inference*, Lecture Notes in Computer Science, pages 402–418, Cham, 2020. Springer International Publishing.
- [Jea67] Piaget Jean. *Childs Conception Of Space*. W. W. Norton and Company, Inc., May 1967.
- [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.
- [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].