An Equational and Graphical Fixed-Point Calculus (Functional Pearl)

GUSTAVO DE MENDONÇA FREIRE, Universidade Federal do Rio de Janeiro, Brazil
HUGO MUSSO GUALANDI, Universidade Federal do Rio de Janeiro, Brazil
HUGO NOBREGA, Universidade Federal do Rio de Janeiro, Brazil
JOAO PAIXAO, Universidade Federal do Rio de Janeiro, Brazil

The fixed-point calculus is a toolbox of theorems for reasoning equationally about fixed points. However, the underlying concepts of the calculus are not defined equationally, including the central definition, that of least fixed point. Thus, although the key theorems of the fixed-point calculus are equational, their proofs are not.

In this work, we give equational specifications for the main concepts of the fixed-point calculus for partially ordered sets, to allow more proofs to be written in an equational style. Since reasoning equationally can be cumbersome without appropriate abstractions, we employ the graphical language of string diagrams, based on a category of feasibility relations, to make reasoning more ergonomical.

Our contributions culminate in graphical equational proofs of key results of the fixed-point calculus, including the rolling, diagonal, and square rules; fixed-point fusion; and the mutual recursion theorem.

CCS Concepts: • **Theory of computation → Equational logic and rewriting**; Denotational semantics.

Additional Key Words and Phrases: Fixed-point calculus, calculational proofs, graphical proofs, string diagrams

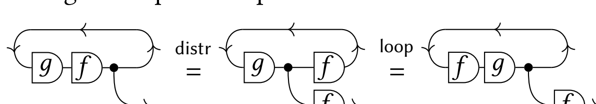
ACM Reference Format:
Gustavo de Mendonça Freire, Hugo Musso Gualandi, Hugo Nobrega, and Joao Paixao. 2026. An Equational and Graphical Fixed-Point Calculus (Functional Pearl). *Proc. ACM Program. Lang.* 10, ICFP, Article 275 (August 2026), 35 pages. https://doi.org/10.1145/3828673

1 Introduction

The *fixed-point calculus* [Aarts et al. 1995; Backhouse 2002] is a toolbox of theorems about fixed points, which model self-referential functions and equations. It has been applied to reason about program derivation [Mu et al. 2009], optimization [Gill and Hutton 2009; Sculthorpe and Hutton 2014], termination [Desharnais et al. 2011], and correctness [Back and von Wright 1999, 2012].

The calculus' toolbox is designed to aid *equational proofs*, comprised of chains of equivalences. Despite that, the toolbox itself had never been proven equationally, until now. Inspired by Piedeleu and Zanasi [2023b], we rebuild the fixed-point calculus on relational foundations and use a graphical syntax of string diagrams to obtain insightful equational proofs for these low-level theorems.

Rolling Rule
$\mu(f \circ g) = f(\mu(g \circ f))$



(1)

Authors' Contact Information: Gustavo de Mendonça Freire, Universidade Federal do Rio de Janeiro, Instituto de Computação, Rio de Janeiro, Brazil, gustavofreire@ic.ufrj.br; Hugo Musso Gualandi, Universidade Federal do Rio de Janeiro, Instituto de Computação, Rio de Janeiro, Brazil, hugomg@ic.ufrj.br; Hugo Nobrega, Universidade Federal do Rio de Janeiro, Instituto de Computação, Rio de Janeiro, Brazil, hugonobrega@ic.ufrj.br; Joao Paixao, Universidade Federal do Rio de Janeiro, Instituto de Computação, Rio de Janeiro, Brazil, jpaixao@ic.ufrj.br.

ACM 2475-1421/2026/8-ART275
https://doi.org/10.1145/3828673

For example, the above is the meat of a graphical proof of the *rolling rule*, which states that if the compositions $f \circ g$ and $g \circ f$ have least fixed points, then the least fixed point of the former is equal to f applied to the least fixed point of the latter. Our proof only uses equalities, whereas standard proofs build on inequalities and implications (see Appendix A). In the diagrams, function application goes from left to right, bifurcations represent copying, and least fixed points are modeled with loops. We start with the diagram for $\mu(f \circ g)$, the least fixed point of $f \circ g$. In the first step, f is duplicated as it distributes over the copy. In the second step, the inner f loops around and the resulting diagram is that of $f(\mu(g \circ f))$, the least fixed point of $g \circ f$ being passed to f.

This concise two-step proof reveals the essence of the rolling rule: that function application distributes over copying. We can be so concise because the diagrammatic proof avoids tedious inequality-manipulating steps that appeal to transitivity, monotonicity, and other basic notions. In fact, there are no inequalities to fiddle with: the notation is free of references to the underlying order relation ($\leq$).

1.1 Equational Foundations

The fixed-point calculus was designed to aid *equational proofs*, that is, proofs based mainly on equational rewriting. So, why have its own theorems not been proven this way already? The main obstacle is that the underlying specification of least fixed points point was not equational.

For technical reasons, the calculus defines least fixed points in terms of the equivalent notion of a least pre-fixed point, which we will use from now on. Let P be a partially-ordered set:

Specification 1.1. A **least pre-fixed point** of a function $f : P \to P$ is an element μf which obeys:

$$f(\mu f) \leq \mu f \qquad \text{Computation}$$

$$\forall x\big(f(x) \leq x \implies \mu f \leq x\big) \qquad \text{Induction}$$

This specification encourages non-equational proofs: to show that some candidate μf is a least prefix point of f, show it is a pre-fixed point (*computation*) and that it is the smallest one (*induction*). The result is a two-part proof that reasons with inequalities and one-way implication.

In Section 2 we reformulate several notions from the theory of partially ordered sets, all in terms of equivalences. Among them, *monotonicity*, *least element*, and, of course, the least pre-fixed point:

Specification 1.2. A **least pre-fixed point** of a function $f : P \to P$ is an element μf which obeys:

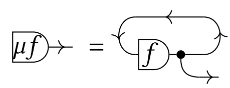
$$\forall y\big(\mu f \leq y \iff \exists x(f\, x \leq x \leq y)\big)$$

In due time we will explain why this second specification is correct. For the current story, the important part is that it encourages an equational proof strategy: to show that some μf is a least pre-fixed point of f, show that the formula on the left can be rewritten into the formula on the right, or vice versa. In Section 2.3 we demonstrate such a proof of the rolling rule.

1.2 Point-Free Graphical Notation

Unfortunately, reasoning equationally in the usual mathematical syntax comes at a cost. As hinted by Specification 1.2, the formulas in the equational proofs have more variables, and their quantifiers are more deeply nested. To abstract those variables away, in Section 3 we turn to point-free reasoning using *string diagrams*.

String diagrams are a formal two-dimensional syntax for morphisms in monoidal categories [Selinger 2011]. Here, the category is that of *feasibility relations* [Censi 2015; Fong and Spivak 2019], a generalization of monotonic functions to a relational setting. Concretely, individual diagrams represent a relation, such as the predicate $P(x) = \mu f \leq x$, and equality of diagrams is relational equality. Thus, the if-and-only-if of Specification 1.2 becomes an equation between diagrams:



In Section 3.1 we describe how to assign meaning to these diagrams and in Section 3.3 we provide a rich set of axioms to manipulate them, which paves the way for fully graphical proofs.

1.3 Lower-Level Abstractions

The sales pitch of the fixed-point calculus was that theorems such as the rolling rule provide equational abstractions over a non-equational foundation. We argue that it can be beneficial to bring equational reasoning down to the level of partially ordered sets.

First, we can equationally prove theorems that previously could not. In Section 4, we graphically state and prove the rolling rule, the diagonal rule, the square rule, and the fixed-point fusion rule. All of these proofs are given in a purely equational style, with the exception of fusion, in which two non-equational steps remain.

Second, our proof technique enables more elementary proofs. For instance, in Section 4.7 we prove the mutual recursion theorem using only fixed-point fusion and fundamental properties of partially ordered sets. In contrast, the existing equational proof is an ingenious one that calls for the square rule, the diagonal rule (three times), and two helper lemmas [Backhouse 2002]. As far as we can tell, our proofs of the mutual recursion theorem and the square rule are the first to be built around fusion of the copy morphism.

Third, equational proofs can deliver fresh insight. For example, our proofs reveal that several important results of the fixed-point calculus reflect properties of the copy operation.

Rolling rule – Monotone functions distribute over copying
Diagonal rule – Copying is associative
Square rule – Copying is commutative

1.4 Limitations

Are equational proofs always better? It depends. We have identified two main limitations to our diagram-based equational proof style (Section 5). The first concerns what kinds of formulas can be represented by the diagrams, and the second comes from "asymmetrical" existence conditions.

The building blocks we use to construct diagrams can describe formulas with existential quantifiers, conjunction, inequalities, and monotonic function application — a fragment of first-order logic called *regular logic* [Butz 1998]. We can also use equality (=) and inclusion (⊆) of diagrams to represent a single bi-implication or implication at the very top of the formula. These diagrams and reasoning rules can describe and prove all of the theorems we present, but they would struggle with problem statements that ask for negations or nested implications.

The other concern is that fully equational proofs are composed of if-and-only-if steps. This saves time when we want to prove both implications at once, but is unsuitable if one implication is valid but its reciprocal is not. We identify a class of *symmetrical* fixed-point problems which fall under the first category, as well as *asymmetrical* problems which fall under the second, where we would need to resort to non-equational ⊆-rules. The asymmetries we encountered occurred in problems that mention some function that does not have a least fixed point. Hence, asymmetry should be less of a problem in settings where the functions of interest are known to have least fixed points.

1.5 Key Contributions

- We provide equational specifications for partially ordered sets and least pre-fixed points, which enable equational proofs for fixed-point theorems (Section 2).
- We show that, with suitable abstractions and with a graphical notation (Section 3), the equational proofs are short and insightful (Section 4). We expose that several existing fixed-point theorems actually reflect properties of the underlying copy operation.
- We identify a class of fixed-point problems that are specially amenable to equational reasoning, as well as classes of problems that are incompatible with it (Section 5).

2 Posets and Fixed Points, Equationally

In this section we provide equational definitions of basic notions from order theory that are relevant to the fixed-point calculus, namely: partially ordered sets, monotonic functions, the least element of a set, least fixed points, joins, and meets. However, before we do so it behooves us to recapitulate the familiar non-equational versions.

Definition 2.1. A **preorder** is a reflexive ($x \leq x$) and transitive ($x \leq y \leq z \implies x \leq z$) relation.

Definition 2.2. A **partial order** is a preorder that is antisymmetric ($x \leq y \wedge y \leq x \implies x = y$).

Definition 2.3. A **partially ordered set**, or **poset**, is a set equipped with a partial order.

Definition 2.4. A **monotonic function** preserves the order of its inputs ($x \leq y \implies f\,x \leq f\,y$).

In the following, let $(P, \leq)$ be a poset, X be a subset of P, and $f : P \to P$ be a monotonic function.

Definition 2.5. A **pre-fixed point** of f is an input x such that $f\,x \leq x$.

Definition 2.6. A **fixed point** of f is an input x such that $f\,x = x$.

Definition 2.7. A **lower bound** of X is a value m that is less than or equal to all elements of X.

Definition 2.8. The **least element** of X is a member of the set that is also a lower bound.

$$m \in X \;\wedge\; \forall x (x \in X \implies m \leq x)$$

Several concepts can be understood as least or greatest elements of some set:

- The **least pre-fixed point** of a function f — written μf.
- The **least upper bound**, or **join**, of a pair of elements a and b — written $a \sqcup b$.
- The **greatest lower bound**, or **meet**, of a pair of elements a and b — written $a \sqcap b$.

The standard specification of least pre-fixed points (1.1) is an instance of the above definition of least element: the computation rule specifies that μf is a pre-fixed point of f and the induction rule specifies that it is a lower bound of the set of pre-fixed points.

A well known result states that the least pre-fixed point is also the least fixed point. For this reason, the two concepts are often used interchangeably. Some problems may ask for the least fixed point, but their proofs might refer to the least pre-fixed point, which is easier to work with.

$$\begin{aligned}
& f(\mu f) \leq \mu f \\
\implies & \quad \{\ f \text{ is monotonic }\} \\
& f(f(\mu f)) \leq f(\mu f) \\
\implies & \quad \{\ \mu f \text{ is the least pre-fixed point }\} \\
& \mu f \leq f(\mu f) \\
\implies & \quad \{\ \text{antisymmetry with } f(\mu f) \leq \mu f \ \} \\
& f(\mu f) = \mu f
\end{aligned}$$

2.1 Posets and Monotonic Functions

The standard definitions for transitivity, antisymmetry, and monotonicity are unfortunately not ideal for equational reasoning. Consider, for instance, how antisymmetry and monotonicity are specified by one-way implications, and furthermore how antisymmetry invites two-part proofs by mutual inclusion. We would prefer to specify everything using equivalences.

Note that all definitions and theorems about any poset $(P, \leq)$ can also be equivalently stated about the opposite poset $(P, \geq)$; we may at times implicitly use the versions for $\geq$ without further mention.

Reflexivity and Transitivity. It is possible to package reflexivity and transitivity into a two-way implication known as the principle of *indirect inequality* [Dijkstra 1991]. This important theorem of equational proof folklore rewrites $x \leq y$ into a form that does not directly compare x to y.

Theorem 2.9 (Indirect Inequality). *A relation $\leq$ is a preorder if and only if*

$$x \leq y \iff \forall z (y \leq z \implies x \leq z) \, .$$

Proof. Refer to Appendix C for detailed proofs of all theorems from Section 2.

Antisymmetry. Typical proofs by antisymmetry work by mutual inclusion: to prove that $x = y$, show that $x \leq y$ and $y \leq x$. The principle of *indirect equality* offers an equational alternative: to prove that $x = y$, show that they compare the same against any other element.

Theorem 2.10 (Indirect Equality). *A preorder is antisymmetric if and only if*

$$x = y \iff \forall z (x \leq z \iff y \leq z) \, .$$

Monotonicity. A function is order preserving if smaller inputs have smaller outputs and greater inputs have greater outputs. Traditionally this is written $x \leq y \implies f\,x \leq f\,y$. We can avoid that one-way implication if we compare $f\,x$ and $f\,y$ indirectly: if we know that $f\,y$ is less than some z, then if we apply f to an input that is even less than y, the result should be even less than z.

Theorem 2.11 (Indirect Monotonicity). *A function f between preordered sets is monotonic iff*

$$\exists y (x \leq y \land f\,y \leq z) \iff f\,x \leq z \, .$$

Several useful corollaries follow from indirect monotonicity with suitable choices of f. For example, indirect monotonicity of the identity function $f(x) = x$ inserts or removes an existential variable in the middle of a chain of inequalities. Indirect monotonicity of the copying function $f(x) = (x, x)$ manipulates chains of inequalities that have the same endpoint.

Notation. We stack formulas side-by-side or above each other as shorthand for conjunctions.

Corollary 2.12. *Given a preordered set $(P, \leq)$, for all $x, z \in P$,*

$$\exists y (x \leq y \quad y \leq z) \iff x \leq z$$

Corollary 2.13. *Given a preordered set $(P, \leq)$, for all $x, z_1, z_2 \in P$,*

$$\exists y \left(x \leq y \quad \begin{array}{c} y \leq z_1 \\ y \leq z_2 \end{array} \right) \iff \begin{array}{c} x \leq z_1 \\ x \leq z_2 \end{array}$$

2.2 Least Elements

The standard specification of the least element compares it against other elements of the set: the least element should belong to the set and be less than the other elements. In contrast, the equational specification we use compares the least element against elements of the universe set. To say that m is the least element of X is to say that an arbitrary y is greater than m if and only if we can find an element of X that is smaller than y.

Theorem 2.14. *m is the **least element** of the poset X if and only if, for all y,*

$$m \leq y \iff \exists x (x \in X \wedge x \leq y)\,.$$

Proof. It can be shown that the $\Leftarrow$ direction specifies that m is a lower bound. Under that assumption, the $\Rightarrow$ direction specifies that m belongs to the set. For the details, see Appendix C. □

Finally, from this specification of least element we can derive equational specifications for least pre-fixed points, least upper bounds, and greatest lower bounds. Let P be a partially ordered set:

Theorem 2.15. *A monotonic $f : P \to P$ has a **least pre-fixed point** $\mu f \in P$ if and only if, for all y,*

$$\mu f \leq y \iff \exists x (f\,x \leq x \leq y)\,.$$

Theorem 2.16. *The **join** of a and b, written $a \sqcup b$, obeys, for all y,*

$$(a \sqcup b) \leq y \iff \begin{array}{l} a \leq y \\ b \leq y \end{array}$$

Theorem 2.17. *The **meet** of a and b, written $a \sqcap b$, obeys, for all y,*

$$\begin{array}{l} y \leq a \\ y \leq b \end{array} \iff y \leq (a \sqcap b)$$

Proof. The least pre-fixed point is the least element of the set $X = \{\, x \mid f\,x \leq x \,\}$. For the join we pick $m = (a \sqcup b)$ and $X = \{\, x \mid a \leq x \wedge b \leq x \,\}$ and then simplify using Corollary 2.13. The meet works the same but with all the inequalities flipped around. □

Since these specifications describe necessary and sufficient conditions, they are of use both when we know that the least element exists and when we do not. Consider, for instance, the specification of the least pre-fixed point (2.15). If we know, or assume, that a function has a least pre-fixed point, the specification tells us about its properties. Conversely, if we do not know that the function has a least pre-fixed point, we can show that it has one by presenting some value that satisfies the properties of the least pre-fixed point.

2.3 An Equational Proof of the Rolling Rule

Now that we have reformulated the specifications of posets, monotonic functions, and least pre-fixed points, we are ready to write equational proofs with them. We will demonstrate a complete proof of the rolling rule. Using equational reasoning, we can transform the fixed point $\mu(f \circ g)$ one step at a time until we arrive at $f\,\mu(g \circ f)$.

Calculation 2.18. Let $f : B \to A$ and $g : A \to B$ be monotonic functions over posets and let $y \in A$.

$$\begin{array}{ll} & \mu(f \circ g) \leq y \qquad (2a) \\ \iff & \quad \{\ \text{Assume } f \circ g \text{ has a least pre-fixed point}\ \} \\ & \exists a_1 \big(f\,(g\,a_1) \leq a_1 \leq y \big) \qquad (2b) \\ \iff & \quad \{\ f \text{ is monotonic}\ \} \end{array}$$

$$
\begin{aligned}
& \exists a_1 \exists b_1 (\, g\, a_1 \le b_1 \quad f\, b_1 \le a_1 \le y \,) && \text{(2c)} \\
\iff & \quad \{\ g \text{ is monotonic }\ \} \\
& \exists a_1 \exists a_2 \exists b_1 (\, a_1 \le a_2 \quad g\, a_2 \le b_1 \quad f\, b_1 \le a_1 \le y \,) && \text{(2d)} \\
\iff & \quad \{\ \text{Corollary 2.13}\ \} \\
& \exists a_2 \exists b_1 (\, f\, b_1 \le a_2 \quad g\, a_2 \le b_1 \quad f\, b_1 \le y \,) && \text{(2e)} \\
\iff & \quad \{\ g \text{ is monotonic }\ \} \\
& \exists b_1 (\, g\,(f\, b_1) \le b_1 \quad f\, b_1 \le y \,) && \text{(2f)} \\
\iff & \quad \{\ f \text{ is monotonic }\ \} \\
& \exists b_1 \exists b_2 (\, g\,(f\, b_1) \le b_1 \le b_2 \quad f\, b_2 \le y \,) && \text{(2g)} \\
\iff & \quad \{\ \text{Assume } g \circ f \text{ has a least pre-fixed point }\ \} \\
& \exists b_2 (\, \mu(g \circ f) \le b_2 \quad f\, b_2 \le y \,) && \text{(2h)} \\
\iff & \quad \{\ f \text{ is monotonic }\ \} \\
& f\, \mu(g \circ f) \le y && \text{(2i)}
\end{aligned}
$$

The proof consists of a series of if-and-only-if steps. Pay attention to how each step introduces or eliminates an existential variable; that indicates where the action is happening and also tells whether the lemma was invoked in a forwards or backwards direction. We begin by invoking the definition of least pre-fixed point (2.15) to transform $\mu(f \circ g) \le y$ into a formula that doesn't mention the μ operator. Next, we use indirect monotonicity (2.11) twice to set up the key step, wherein Corollary 2.13 duplicates the f. After more applications of indirect monotonicity, we employ the definition of least pre-fixed point to express $\mu(g \circ f)$ and ultimately arrive at $f\, \mu(g \circ f)$.

Summing up, from this calculation we can extract two versions of the rolling rule. The simplest one to state is applicable when both fixed points are known to exist [Backhouse 2002].

Theorem 2.19. *If both $f \circ g$ and $g \circ f$ have a least pre-fixed point, then $\mu(f \circ g) = f\, \mu(g \circ f)$.*

Proof. Indirect equality (2.10) using steps 2a–2i. □

The stronger version only assumes that $g \circ f$ has a least pre-fixed point [Backhouse 2021].

Theorem 2.20. *If $g \circ f$ has a least pre-fixed point, then $f \circ g$ also does and $\mu(f \circ g) = f\, \mu(g \circ f)$.*

Proof. Specification of least pre-fixed point (2.15) using steps 2b–2i. □

Unfortunately, the converse does not hold. Steps 2a–2f seem to invite a proof that if $f \circ g$ has a least fixed point then so does $g \circ f$, but the $f\, b_1$ in step 2f gets in the way.

3 Feasibility Relations and Their String Diagrams

In the previous section's proof of the rolling rule, over half of the steps were busywork with indirect monotonicity, and we were buried under a pile of existential variables. String diagrams address both of these problems. In the new notation, formulas that are equivalent up to monotonicity have the same diagram, avoiding the need for monotonicity steps. The diagrams are also point-free: there are no quantifiers and no variables for the elements of the underlying preorders; the only variables that appear are for maps and relations. As a result, the proofs become shorter and more readable.

The first step towards the graphical notation is to note that the propositions that appear in Calculation 2.18 may be understood as relations between their free variables (the ones that are not existentially quantified). Because only monotonic functions are used, the propositions still hold if

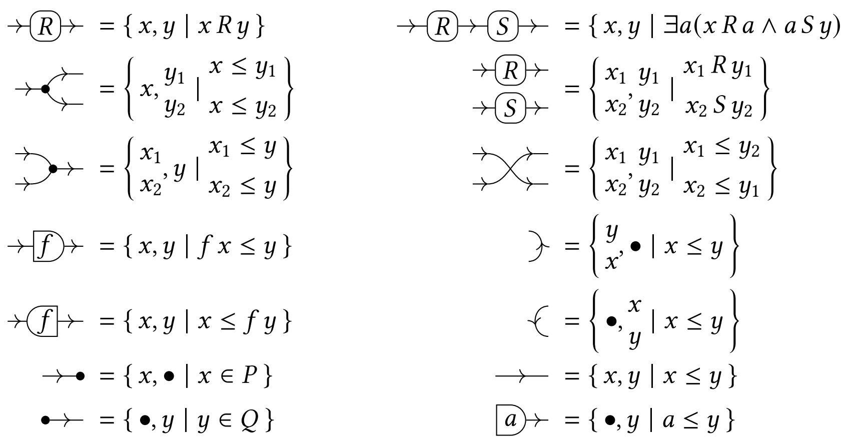


Fig. 1. Diagram generators and their feasibility relations. x and y respectively refer to inputs and outputs. P and Q are the input and output posets. The unit element $\bullet$ is the single element in the unitary poset I. Vertically stacked variables are tuples and vertically stacked formulas are conjunctions.

we decrease a free variable that appears to the left of a $\le$ or increase a free variable that appears to the right of a $\le$. This kind of relation is known as a *feasibility relation* [Censi 2015].

Definition 3.1. A feasibility relation between preorders $(P, \le_P)$ and $(Q, \le_Q)$ is a relation $R \subseteq P \times Q$ between their underlying sets satisfying

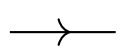
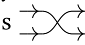
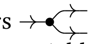
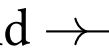
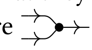
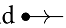
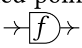
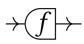
$$\begin{aligned} x_1 \le_P x_2 \wedge x_2\, R\, y &\implies x_1\, R\, y \\ x\, R\, y_1 \wedge y_1 \le_Q y_2 &\implies x\, R\, y_2\,. \end{aligned}$$

Definition 3.2. The category *Feas* has preorders as objects and feasibility relations as arrows, with the identity at $(P, \le_P)$ given by $(\le_P) = \{ x, y \mid x \le_P y \}$. With monoidal product taken to be the categorical product of preorders, it forms a symmetric monoidal category, and, considering relational inclusion as a partial order on each hom-set, *Feas* is order-enriched compatibly with the monoidal product.

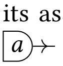
For a more complete discussion of the concept, the reader may consult Fong and Spivak [2019]. The category *Feas* bakes the ordering relations into the structure of the category itself and its identity morphisms. In *Feas*, we do not explicitly refer to $\le$ and thus we avoid having to fiddle with transitivity and monotonicity to shuffle those inequalities around. This declutters the notation and facilitates the search for an equational proof, as should become apparent when we discuss the proofs in Section 4.

3.1 Graphical Notation

As we saw in our proof of the rolling rule, complex feasibility relations that make extensive use of relational composition can become hard to read. In response to that, we employ a graphical point-free style based on the syntax of string diagrams. The fundamental diagrammatic building blocks, called generators, are outlined in Figure 1.

Each diagram corresponds to a feasibility relation comparing elements from posets, with the posets themselves left implicit in the notation. Connecting two diagrams horizontally composes their relations sequentially, and stacking two diagrams vertically composes them in parallel. The diagram  is the identity of sequential composition and the empty relation is the identity of parallel composition, denoting conjunction of relations. Crossed wires  allow inputs and outputs to be reordered at will. The black node operators  and  respectively copy and discard input variables, in the sense that they make those variables be used more than once or not at all. The duals of those diagrams are  and , which copy and discard output variables. Diagrams  and  turn outputs into inputs, and allow for the diagrammatic representation of fixed-point loops. Finally, a monotone map f gives rise to a feasibility relation in two ways, depicted as  and .

Wires are annotated with a small arrow which indicates the direction of the poset comparisons. More specifically, following the convention of Piedeleu and Zanasi [2023a], when a composite diagram uses graphical generators for both $(P, \leq)$ and $(P, \geq)$, those for $(P, \leq)$ are drawn with arrows pointing to the right while those for $(P, \geq)$ get arrows pointing left, helping avoid confusion. Mnemonically, elements get bigger in the direction of the arrow.

Since the elements $a \in P$ of some poset are in one-to-one correspondence with monotone mappings from the unitary poset $I = \{\bullet\}$ to P, we will reuse the name of the element a for its associated mapping $a : I \to P$, allowing us to express graphically the feasibility relation  $= \{ \bullet, y \mid a \leq y \}$, of type $I \to P$ in *Feas*, which corresponds to the set of elements greater than or equal to a. We also remark that the associators and unitors of *Feas*, responsible for conversions such as $(x, (y, z)) \mapsto ((x, y), z)$ and $(\bullet, x) \mapsto x$, although formally required, are not represented graphically.

3.2 The Diagram for a Least Fixed Point

For a concrete example, let's derive the formula for the diagram in Figure 2. It appeared in the rolling rule and represents the set of all values that are greater than some pre-fixed point of $f \circ g$.

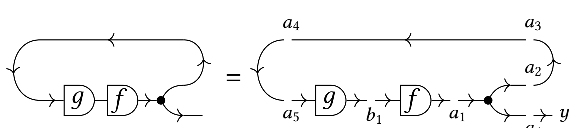
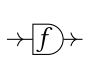


Fig. 2. The diagram for the set of values that are larger than or equal to some pre-fixed point of $f \circ g$.

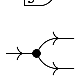
In it, there are no loose ends on the left, and there is a single loose end to the right, which we will label y. Formally, this relation is a set of pairs of the form $(\bullet, y)$ where $\bullet$ is the element of the unitary poset I. In practice, it behaves as a predicate with a single parameter y. Next, we assign meaning to the body of the diagram. We break it down into its constituent generators, each of which introduces one or two inequalities.

 $= \{ a_5, b_1 \mid g\, a_5 \leq b_1 \}$

 $= \left\{ \begin{matrix} a_3 \\ a_2 \end{matrix}, \bullet \mid a_2 \leq a_3 \right\}$

 $= \{ b_1, a_1 \mid f\, b_1 \leq a_1 \}$

 $= \{ a_4, a_3 \mid a_3 \leq a_4 \}$

 $= \left\{ a_1, \begin{matrix} a_2 \\ a_6 \end{matrix} \;\middle|\; \begin{matrix} a_1 \leq a_2 \\ a_1 \leq a_6 \end{matrix} \right\}$

 $= \left\{ \bullet, \begin{matrix} a_4 \\ a_5 \end{matrix} \mid a_4 \leq a_5 \right\}$

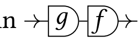
 $= \{ a_6, y \mid a_6 \leq y \}$

Relational composition collects all these inequalities and turns the connection points into existential variables:

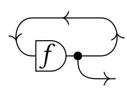
$$\left\{ \bullet, y \mid \exists a_1 \cdots a_6 b_1 \begin{pmatrix} a_1 \leq a_2 \leq a_3 \leq a_4 \leq a_5 \\ g\, a_5 \leq b_1 \quad f\, b_1 \leq a_1 \leq a_6 \leq y \end{pmatrix} \right\} \tag{3}$$

Through repeated applications of indirect monotonicity we can eliminate all variables except a_1. This reveals that the predicate accepts all y that are greater than some pre-fixed point of $f \circ g$.

$$\{ \bullet, y \mid \exists a_1 \big((f \circ g)\, a_1 \leq a_1 \leq y \big) \} \tag{4}$$

As a rule of thumb, each loose wire becomes a parameter to the relation and each internal wire becomes an existentially-quantified variable. Adjacent function generators, as in  , can be collapsed into a function composition without an existential variable in between.

In the general case, in which the loop body is taken to consist of a single function f, an analogous process would show that



$$= \{ \bullet, y \mid \exists a_1 \big(f\, a_1 \leq a_1 \leq y \big) \}\,. \tag{5}$$

By the specification of least pre-fixed points (2.15), this set has $(\bullet, y)$ as an element exactly when $\mu f \leq y$, as long as f admits a least pre-fixed μf. Conversely, showing that there exists some μf for which this equivalence holds regardless of the choice of y implies that it must be the least pre-fixed of f. This insight bridges the gap between the study of least fixed points and the graphical calculus of *Feas*, and is what enables contributions such as the ones in [Piedeleu and Zanasi 2023a] and in the present work.

There are a couple of benefits that come from reasoning about μf in this indirect style, through the relation above. The first is that, unlike the approach symbolized by Specification 1.1, which studies the element μf as atomic, the diagram in (5) is naturally decomposed into simpler parts, with the relation as a whole expressed compositionally. Such compositionality is made possible by working with relations. In particular, it is often the case that graphical proofs are built around properties of the copy morphism, which appears in the diagram, but that is not present in Specification 1.1. A second benefit is that, unlike μf, which may not exist for a given monotonic function f in a general poset, the diagram on the left depicts a relation which is always well-defined. This means that many theorems can be phrased in terms of relational equalities, and, in this form, hold unconditionally. Then, introducing some existence-related hypotheses allows the original formulations, in terms of the element μf, to be recovered.

3.3 Rules for Manipulating Diagrams

For the graphical proofs, we will need rules with which to manipulate diagrams. We list them in Figure 3, which is divided into five blocks. Most of the rules are equations, but a handful are inequations ($\subseteq$). Although we are primarily interested in equational proofs, the system also allows non-equational reasoning.

- The A rules state the basic properties of parallel and sequential composition, as well as of the twist operation.
- The B rules state that copy is associative, commutative, and has discard as its identity (B1–B6). Moreover, they govern how relations distribute over copy and interact with discard (B7–B10), and also state that horizontal and vertical composition are compatible with the ordering on relations (B11-B12).

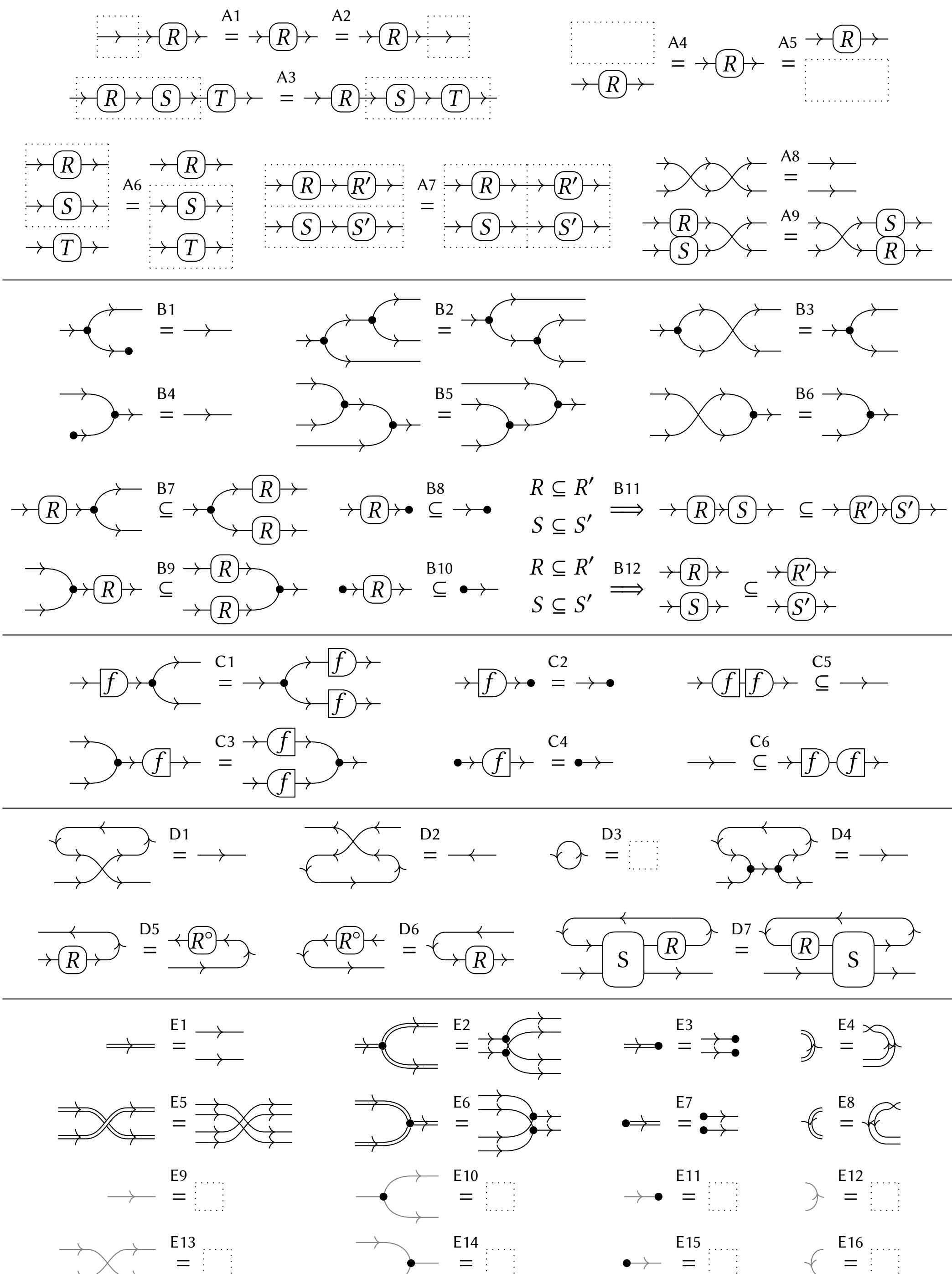


Fig. 3. Rules for rewriting diagrams

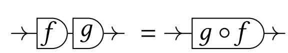
- The C rules state how monotone functions distribute over copies and interact with discards (C1–C4), and how the sequential compositions of monotone functions and their duals compare with the identity relation (C5–C6).
- The D rules describe properties of the and generators; there are rules to eliminate trivial loops (D1–D4) and rules to push relations around a loop (D5–D6). The notation R^o stands for the *opposite* of R, relating precisely the pairs x, y for which R relates y, x. Rule D7 pushes a relation all the way around, which happens if we apply D5 and D6 in quick succession.
- The E rules concern product posets $P \times Q$ and the unitary poset $I = \{\bullet\}$. Double lines represent wires in the product poset, and the rules state that they may be broken down into parallel diagrams in the respective posets. Gray lines represent wires in the unitary poset, and the rules state that they may be deleted.

This set of axioms is not intended to be minimal. For example, rules (C1–C4) can be derived from (B7–B10) and (C5–C6).

The correctness of the rules is simple to prove using the equational lemmas developed in Section 2, and to make this point evident, we will exhibit some proofs in this style. The remaining proofs may be found in Appendix E, together with pointful translations of all the rules.

The following is a lemma that we will apply implicitly when drawing and interpreting diagrams. The relation on the left tends to be more useful in graphical proofs, because it allows f and g to be separately manipulated.

Lemma 3.3. *For all monotone $f : P \longrightarrow Q$ and monotone $g : Q \longrightarrow R$,*

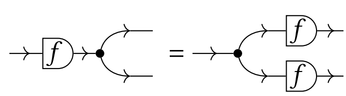


Proof. If we label the incoming wire as w and the outgoing wire as z, the equivalence we seek is revealed to be a special case of indirect monotonicity of g, with $x = f\,w$.

$$\exists y (f\,w \le y \wedge g\,y \le z) \iff g\,(f\,w) \le z. \qquad \square$$

The next lemma restates rule (C1), which says that monotone functions distribute over the copy.

Lemma 3.4. *For all monotone $f : A \longrightarrow B$,*



Proof. This follows from indirect monotonicity (2.11) and its corollaries (2.13).

$$\exists b \begin{pmatrix} f\,x \le b & \begin{matrix} b \le y_1 \\ b \le y_2 \end{matrix} \end{pmatrix} \overset{(2.13)}{\iff} \begin{matrix} f\,x \le y_1 \\ f\,x \le y_2 \end{matrix} \overset{(2.11)}{\iff} \exists a_1 a_2 \begin{pmatrix} x \le a_1 & f\,a_1 \le y_1 \\ x \le a_2 & f\,a_2 \le y_2 \end{pmatrix} \qquad \square$$

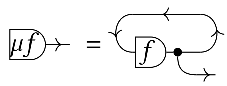
4 The Fixed Point Calculus, Graphically

Armed with the point-free diagrammatic syntax, we may turn our attention to the theorems of the fixed-point calculus. The first step is to translate the reasoning principles of Section 2 into this new language. Indirect monotonicity (2.11) is already covered by the identity of composition in *Feas* (A1, A2). Indirect inequality (2.9), indirect equality (2.10), and the specification of least pre-fixed points (2.15), can be represented graphically as follows:

Theorem 4.1 (Graphical Indirect Inequality). $a \le b \iff$ 

Theorem 4.2 (Graphical Indirect Equality). $a = b \iff$ 

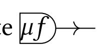
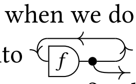
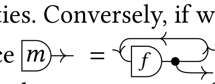
Theorem 4.3 (Graphical Least Fixed Points). *A monotonic $f : P \to P$ has a least pre-fixed point $\mu f \in P$ if and only if*



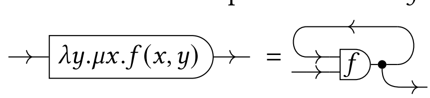
Some theorems in the upcoming section mention fixed points of large lambda expressions. To make them easier to read, we introduce a special notation for these expressions.

Notation. We write $\mu x.E$ as a shorthand for $\mu(\lambda x.E)$.
Thus, $\mu x.f(x, y)$ is shorthand for $\mu(\lambda x.f(x, y))$.

Indirect equality allows us to move between equalities at the poset level and relational equivalences in string diagrams. Thus, if we want to prove some equality we can restate the problem in terms of string diagrams and do the rest of the proof that way. Similarly, indirect inequality allows us to transit between inequalities and relational inclusion.

Like its pointful counterpart, the graphical specification of least pre-fixed points (4.3) can be used both when we know the function has a least pre-fixed point exists and when we do not know. If we know that f has a least pre-fixed point, then we can rewrite  into , enabling further rewrite opportunities. Conversely, if we want to show that f has a least pre-fixed point, we should seek an equivalence , which tells that m is the least pre-fixed point of f.

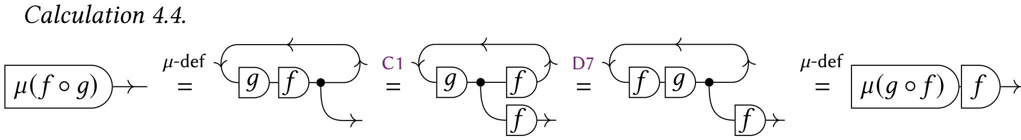
Lastly, we should discuss how to represent fixed points involving functions of multiple variables. Consider a multi-variable function $f : (P \times Q) \to P$. If we were to fix the second parameter, this function would become a single-variable function from P to P. Thus, $\lambda y.\mu x.f(x, y)$, is the function that receives y and returns the least fixed point of $\lambda x.f(x, y)$. Graphically, this fixed point is represented by a diagram with an additional input wire for the y.

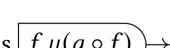


4.1 Rolling Rule

Having developed the syntax and the properties of the graphical notation for feasibility relations, we return to the rolling rule, which we studied in a pointful style in Section 2.3. Once again, we start by calculating, but now diagrammatically, seeking to understand how least fixed points interact with function composition.

Calculation 4.4.



Recall, per Lemma 3.3, that the last diagram could have been written as  .

Calculation 4.4 is analogous to Calculation 2.18, but is expressed in the diagrammatic notation. The benefits in readability are undeniable: the tedious indirect monotonicity steps are gone, as are the existential quantifiers and their variables. The new calculation distills the essence of Theorems 2.19 and 2.20, which is precisely what a great proof should do. The repetition of f in $f\,\mu(g \circ f)$ comes from the interaction between f and the copy morphism, and the "rolling" motion that makes f and g switch places comes from the loop intrinsic to the notion of a fixed point.

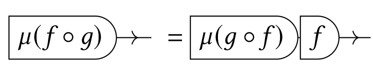
Moreover, the diagrams make clear how one could *discover* the rolling rule. In the traditional style prescribed by Specification 1.1, we would need to creatively guess that $\mu(f \circ g)$ and $f\mu(g \circ f)$

are equal and then check that $f\ \mu(g \circ f)$ satisfies both the induction and computation properties of $\mu(f \circ g)$. The equational style of calculation made possible by Theorem 2.15 allows one to start with $\mu(f \circ g)$ and rewrite step-by-step until arriving at $f\ \mu(g \circ f)$, only then finding out that the equality holds. Complementarily, the diagrammatic notation gives the syntactic clarity required to spot which rewrite possibilities are available, supporting the calculational process of theorem discovery.

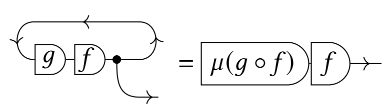
This is in contrast to the pointful language we used in the previous proof of this result. When the propositions were expressed with variables and quantifiers, rewrite opportunities arose when multiple subexpressions referred the same variable, but that could be hard to spot if the expressions were not adjacent in the page. With the switch to the diagrammatic point-free notation, the connected subexpressions now share a wire and are brought together pictorially, turning the calculations into a visual process of pattern-matching and rewriting.

With Calculation 4.4 in hand we can derive several forms of the rolling rule. First we can recover graphical proofs of Theorems 2.19 and 2.20. Let $f : B \longrightarrow A$ and $g : A \longrightarrow B$ be monotonic functions.

(1) Theorem 2.19: if both $f \circ g$ and $g \circ f$ have a least pre-fixed point, then $\mu(f \circ g) = f\ \mu(g \circ f)$. This follows from parts 1–5 of Calculation 4.4 together with graphical indirect equality (4.2).

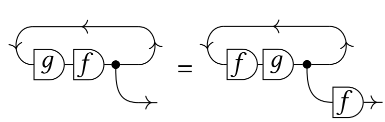


(2) Theorem 2.20: if $g \circ f$ has a least pre-fixed point, then $f \circ g$ also does and $\mu(f \circ g) = f\ \mu(g \circ f)$. This follows from parts 2–5 of Calculation 4.4 together with the specification of least pre-fixed points (4.3).



Lastly, using the graphical syntax we can state a more general version that encapsulates the meat of the proof. It doesn't assume the existence of either least pre-fixed point.

Theorem 4.5 (Graphical Rolling Rule). *For all monotonic $f : B \longrightarrow A$ and $g : A \longrightarrow B$,*



Proof. Parts 2–4 of Calculation 4.4. □

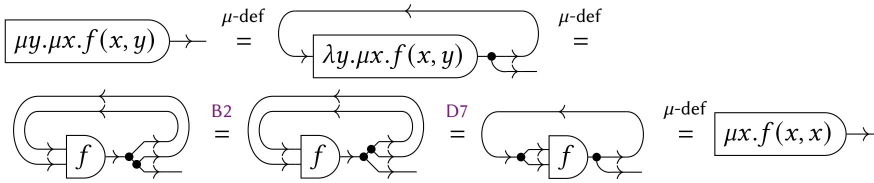
In pointful style this is quite the tongue-twister:

$$\text{For all } y \in A, \quad \exists a\big((f \circ g)\, a \leq a \wedge a \leq y\big) \iff \exists b\big((g \circ f)\, b \leq b \wedge f\, b \leq y\big).$$

4.2 Diagonal Rule

The next theorem we will study diagrammatically is the *diagonal rule*, which analyzes what happens when least fixed points nest, as in $\mu y.\mu x.f(x, y)$, which makes sense for $f : P \times P \longrightarrow P$. Let's jump straight to the calculations.

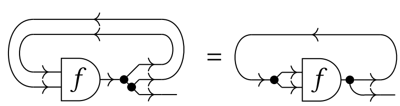
Calculation 4.6. We begin by unpacking the fixed points in $\mu y.\mu x.f(x, y)$. In the first step, the outermost μ becomes a λ. In the second step, we expand the inner fixed point loop. Since y is a free variable in the inner function, this inner loop has an incoming wire from the outer loop. We now reach the key step in the proof, where we invoke associativity of the copy morphism (B2). After that, all that remains is to roll the inner copy to the other end of the loop (D7), revealing $\lambda x. f(x, x)$.



The end result is, at first glance, rather surprising: nested uses of μ may always be denested by merging both variables into one. Reusing this variable as both arguments of f is the same as precomposing it with the *diagonal map* $\Delta(x) = (x, x)$, and this is what gives the theorem its name.

We may now formally state the conclusions of this calculation. As before, the core of the proof is easier to state using graphical language.

Theorem 4.7 (Graphical Diagonal Rule). *Let $f : P \times P \to P$ be a monotonic function. Then,*

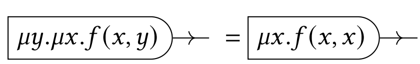


Reading it as a whole, without giving the existence hypothesis much thought, we get the weak formulation, which is more convenient when all the least fixed points are known to exist.

Theorem 4.8. *Let $f : P \times P \to P$ be a monotonic function. If the least fixed point $\mu x. f(x, y)$ exists for all y, and both $\mu y.\mu x. f(x, y)$ and $\mu x. f(x, x)$ also exist, then*

$$\mu y.\mu x. f(x, y) = \mu x. f(x, x).$$

Proof. In Calculation 4.6, apply Indirect Equality to



□

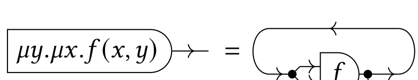
But, with a more careful analysis, we can extract a bit more by noticing that parts of Calculation 4.6 fit the specification of pre-fixed points (4.3). This means that, from this one calculation, we get two existence theorems. However, the subexpression $\mu x. f(x, y)$ has an associated existence hypothesis that is applied in an intermediary step in the calculation, and so this least fixed point cannot have its existence deduced. Thus, for each $y \in P$, the existence of $\mu x. f(x, y)$, i.e, the least pre-fixed point of $\lambda x. f(x, y)$, will always be assumed.

Theorem 4.9 (Diagonal Rule). *Let $f : P \times P \to P$ be a monotonic function such that, for all $y \in P$, $\mu x. f(x, y)$ exists. If either one of $\mu y.\mu x. f(x, y)$ or $\mu x. f(x, x)$ exists, the other one also exists and they are equal. More formally,*

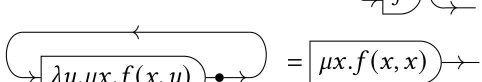
(1) if $\lambda y.\mu x. f(x, y)$ has a least pre-fixed point then $\lambda x. f(x, x)$ also does, and
(2) if $\lambda x. f(x, x)$ has a least pre-fixed point then $\lambda y.\mu x. f(x, y)$ also does,

and, in both cases, we have $\mu y.\mu x. f(x, y) = \mu x. f(x, x)$.

Proof. Parts of Calculation 4.6 fit Theorem 4.3:



proves (1), and



proves (2).

□

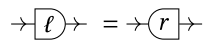
4.3 Galois Connections

In this section we review the well-known concept of a Galois connection, which will be required for the upcoming fixed-point fusion rule. Consider monotone maps $\ell \colon P \to Q$ and $r \colon Q \to P$.

Definition 4.10. The maps ℓ and r form a **Galois connection** when, for all $x \in P$ and $y \in Q$,

$$\ell\, x \leq y \iff x \leq r\, y\,. \tag{G1}$$

Graphically, we may write



(G2)

The mapping ℓ is called the *left adjoint* and r is the *right adjoint*.

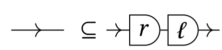
Equivalently, Galois connections can be defined in terms of *cancellation rules*, which govern the composition of ℓ and r.

Theorem 4.11. *Monotone maps ℓ and r form a* ***Galois connection*** *when, for all $x \in P$ and $y \in Q$,*

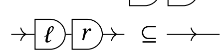
$$(\ell \circ r)\, y \leq y \tag{G3}$$

$$x \leq (r \circ \ell)\, x \tag{G4}$$

Graphically, we may write



(G5)



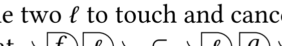
(G6)

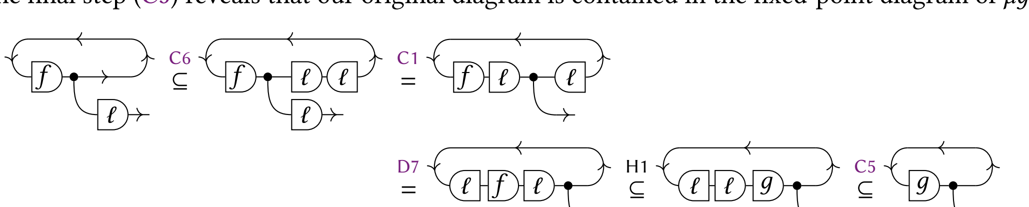
Proof. Refer to Appendix D. □

4.4 Fixed-Point Fusion

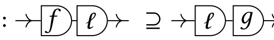
The fixed-point fusion theorem gives conditions under which the application of a monotonic function to a least pre-fixed point is again a least pre-fixed point, as in $\ell\, \mu f = \mu g$. This theorem allows short equational proofs for many results that would otherwise need to be proved via fixed-point induction.

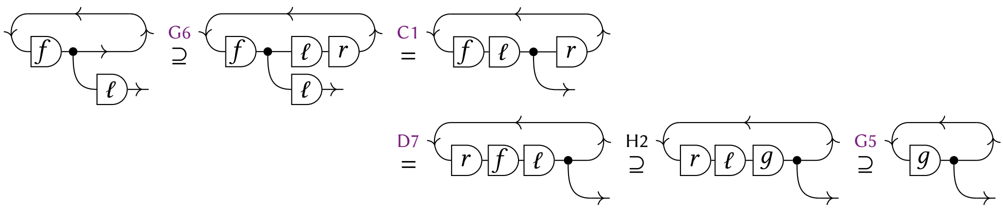
This is the only theorem of the fixed-point calculus that we will prove by non-equational reasoning. We let ℓ, f, and g be monotonic functions and seek a proof that $\ell\, \mu f = \mu g$.

Calculation 4.12. We begin with the fixed-point diagram for $\ell\, \mu f$. From the get-go, there seems to be no way to push ℓ inside the loop, so we must settle for a two-sided proof that drops the equality. One option is to start with (C6) and (C1). After that, we would like the two ℓ to touch and cancel. To make them touch, in (H1) we introduce and apply the hypothesis that  . The final step (C5) reveals that our original diagram is contained in the fixed-point diagram of μg.

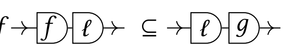


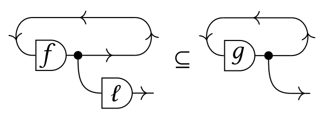
Calculation 4.13. We can still recover an equality if we obtain the reverse inclusion, but this would require us to push ℓ through the copy while making the relation *smaller*. For this, the cancellation

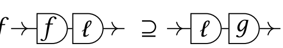
rules for Galois connections come in handy. If we have ℓ left adjoint to some r then we can take a different route, one that starts with (G6) and uses the hypothesis (H2):  .

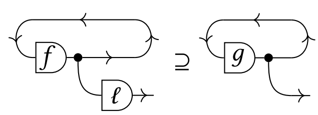


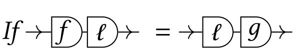
From these calculations, we can collect a few theorems. Let $\ell\colon A \longrightarrow B$, $f\colon A \longrightarrow A$ and $g\colon B \longrightarrow B$ be monotonic functions between partially ordered sets.

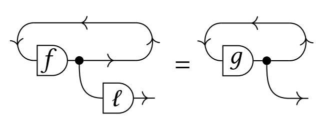
Lemma 4.14. *If*  *, then*

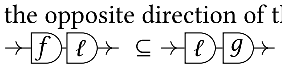


Lemma 4.15. *If*  *and ℓ is a left adjoint, then*



Theorem 4.16 (Graphical μ-fusion). *If*  *and ℓ is a left adjoint, then*

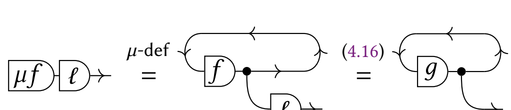


Remark. Be aware that the above is expressed in terms of relational inclusion ($\subseteq$), which points at the opposite direction of the poset comparisons ($\geq$) commonly found in the literature. For instance,  is equivalent to $\forall x\big((\ell \circ f)\, x \geq (g \circ \ell)\, x\big)$, by indirect inequality (2.9).

Theorems 4.14–4.16 don't assume that least pre-fixed points of f and g exist. Similarly to what we experienced with the rolling rule, these theorems are well-suited for graphical reasoning, but are unwieldy in a pointful language. To recover the more traditional statement of the fixed-point fusion rule, we can observe that there is an opportunity to define the least fixed point of g in terms of the least fixed point of f, because the loop around the g doesn't have any functions hanging out of its tail.

Theorem 4.17 (μ-fusion). *Let $\ell\colon A \longrightarrow B$, $f\colon A \longrightarrow A$ and $g\colon B \longrightarrow B$ be monotonic functions satisfying $\ell \circ f = g \circ \ell$, with ℓ being a left adjoint. If f has a least pre-fixed point, then g also does and $\mu g = \ell\, \mu f$.*

Proof. In the first step we use the assumption that μf is the least fixed point of f and in the second we fuse ℓ into the loop (Theorem 4.16). The complete equation shows that μg exists (Theorem 4.3).



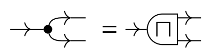
□

The situation with fixed-point fusion is somewhat curious because the inequalities used in Calculations 4.12 and 4.13 can individually be expressed as equivalences: (C5) amounts to transitivity, (C6) amounts to monotonicity, and Galois connections can be described with an equivalence (G2). Nevertheless, we could not find a way to assemble all of them into an equational proof of fusion.

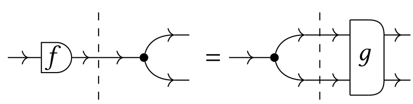
4.5 Fusing the Copy

As evidenced by the Galois connection for the meet (2.17), the copying function $\ell(y) = (y, y)$ is left adjoint to the binary meet function $r(a, b) = a \sqcap b$, which makes it a candidate for fusion. Of course, a necessary and sufficient condition for this is that the meet function must exist and must be total: the poset must have all binary meets. Graphically, we can express this requirement using the Galois connection between the copying function and the binary meet.

Definition 4.18 (All binary meets exist). There exists a function $\sqcap : (P \times P) \to P$ such that

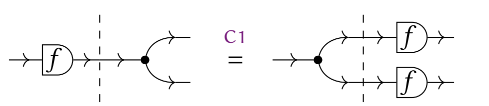


Having established when we can fuse the copy, the next step is to satisfy the fusion hypothesis $\ell \circ f = g \circ \ell$. To do so, we must find suitable functions that can stand in for g in this diagram:

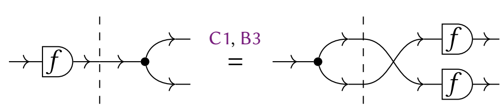


(7)

We will show two such functions. The simplest option comes from rule (C1): we can choose g to be the function that receives two inputs and applies f to each. The second way adds a twist right after the copy, taking advantage of the fact that copies are commutative (B3).



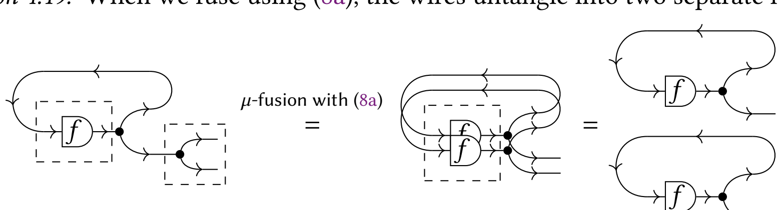
(8a)



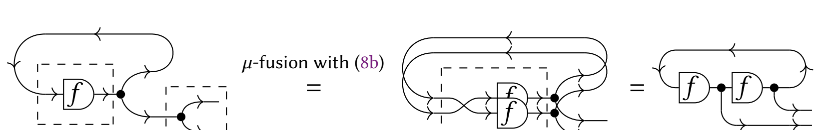
(8b)

Applying fixed-point fusion to these hypotheses initially produces a double loop with two sets of parallel wires, because g takes two inputs and produces two outputs. After that, we can untangle the wires to obtain the final result. The two possibilities are as follows:

Calculation 4.19. When we fuse using (8a), the wires untangle into two separate loops.



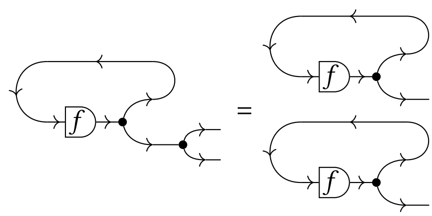
Calculation 4.20. When we fuse using (8b), the result is a single loop with two exits.



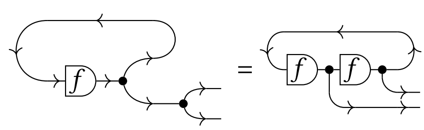
Technically, the above calculations also employ several rules from the E block in Figure 3 to represent the result of fusion as a double loop, as we have done, instead of as a single-wire loop in the product poset $P \times P$.

In summary, copy-fusion takes a fixed-point diagram where f appears once and transforms it into a different diagram where f appears twice.

Lemma 4.21 (Copy fusion). *Let $f : P \to P$ be a monotonic function on a poset P that has all binary meets. We have:*



(9a)



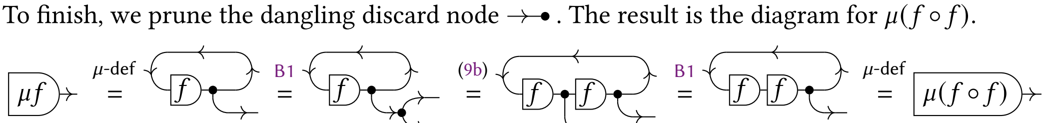
(9b)

In the following sections we will use the two forms of copy-fusion to prove the square rule and to solve mutually-recursive fixed-point equations. The main requirement for copy-fusion is that the poset must have all binary meets or, equivalently, that the poset has meets for every finite non-empty set. Stronger conditions also suffice. For example, copy-fusion is valid for complete lattices or bounded-complete posets.

4.6 Square Rule

The square rule is a theorem that is closely related to the rolling rule. It equates the least fixed points μf and $\mu(f \circ f)$. We can prove this rule by using the second form of copy-fusion, which relies on the facts that functions distribute over copying and that the copying is commutative.

Calculation 4.22. We start with the fixed-point diagram for μf. To kick-start the transformations, we add a copy immediately followed by a discard and then we fuse the copy using hypothesis (9b). To finish, we prune the dangling discard node $\rightarrowtail\!\bullet$. The result is the diagram for $\mu(f \circ f)$.



Looking at different segments of Calculation 4.22 yields a few results. First, reading it as a whole allows us to conclude the following.

Theorem 4.23. *Let P be a poset with all binary meets and $f : P \to P$ a monotonic function. If both f and $f \circ f$ have a least fixed point, then $\mu f = \mu(f \circ f)$.*

Proof. Apply Indirect Equality to the first and the last step in Calculation 4.22.

$$\boxed{\mu f} \rightarrowtail \; = \; \boxed{\mu(f \circ f)} \rightarrowtail$$

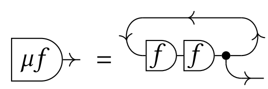
□

However, upon closer inspection, we see that two equivalences that appear in the calculation have the shape required by the specification of pre-fixed points (4.3). This means that we can get two existential implications out of this one calculation.

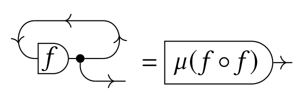
Theorem 4.24 (Square Rule). *Let P be a poset with all binary meets and $f : P \to P$ a monotonic function. If either one of f or $f \circ f$ has a least pre-fixed point, the other one also has one, and they are the same. In other words,*

(1) if f has a least pre-fixed point, then $f \circ f$ also has one and it is the same.
(2) if $f \circ f$ has a least pre-fixed point, then f also has one and it is the same.

Proof. If f has a least pre-fixed point, the equivalence between the first and penultimate steps of Calculation 4.22 shows that it satisfies the characterization of the least pre-fixed point of $f \circ f$ given by Theorem 4.3. Note that this segment of the calculation does assume that $f \circ f$ has a least pre-fixed point.



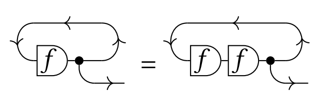
Conversely, if $f \circ f$ admits a least pre-fixed point, the equivalence between the second and the last steps shows that it satisfies the characterization of the least pre-fixed point of f.



□

Still, the more general statement is the one that only looks at the middle part of the calculation.

Theorem 4.25 (Graphical Square Rule). *Let P be a poset with all binary meets and $f : P \to P$ a monotonic function. Then,*



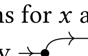
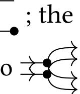
In this section, we presented a novel equational proof of the square rule, using copy-fusion. Our method has the advantage that it proves both directions of Theorem 4.24 in one go. The downside is that Theorem 4.24 is not the strongest form of the square rule. It turns out that direction (1) does not need the assumption that the poset has all binary meets; only direction (2) does [Backhouse 2021]. (For a counterexample, see Appendix B.) Unfortunately, our proofs are built from symmetrical proof steps, which cannot prove asymmetrical propositions. We will return to this matter in Section 5.2.

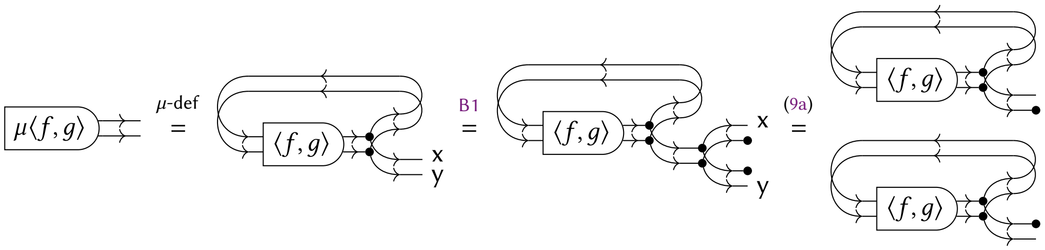
4.7 Mutual Recursion

To complete our tour of the fixed-point calculus, we will discuss the theorem that solves mutually recursive fixed-point equations one variable at a time. Suppose that we want to find the least solution of a two-by-two system of inequalities:

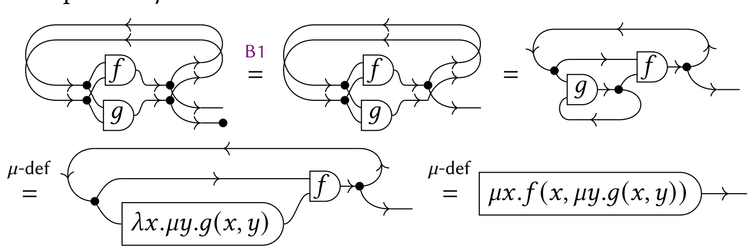
$$x \geq f(x, y)$$
$$y \geq g(x, y)$$

That is, given $f : P \times Q \to P$ and $g : P \times Q \to Q$, we seek the least pre-fixed point of the function $\langle f, g \rangle : P \times Q \to P \times Q$.

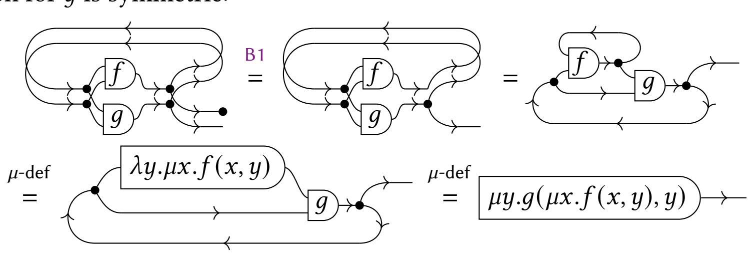
Calculation 4.26. We begin with a fixed-point loop for $\langle f, g \rangle$, which has two sets of wires and two outputs, one for x and one for y. We aim to separate the equations for x and y and for that we use rule (B1) twice: one time we copy x and discard the bottom copy ; the other time we copy y and discard the top copy . These side-by-side copies give rise to , a copy in the product poset $P \times Q$, which may be fused. If we fuse using (9a), we end up with two disjoint loops that each solve for a single variable. The top loop solves for x and the bottom one solves for y.



All that is left is to solve the fixed point equations for x and y. To solve for x, we first expand the definition of $\langle f, g\rangle$ and prune the discarded y output (B1). The y becomes internal to the calculation and we can rearrange the nodes and wires to emphasize that. (We invite the reader to verify that after the second step each wire is still connected to the same place.) Finally, we convert the diagram to a least pre-fixed point in μ notation.



The solution for y is symmetric:



From these calculations, we may extract our final theorem.

Theorem 4.27 (Mutual Recursion). *Let P and Q be posets with all binary meets and let the functions $f : P \times Q \to P$ and $g : P \times Q \to Q$ be monotonic functions. If all the least pre-fixed points mentioned in the right hand side exist, then $\langle f, g\rangle$ has a least pre-fixed point given by*

$$\mu\langle f, g\rangle = (x_0, y_0) \quad \textit{where} \quad \begin{aligned} x_0 &= \mu x.f(x, \mu y.g(x, y)) \\ y_0 &= \mu y.g(\mu x.f(x, y), y) \end{aligned}$$

To conclude, we used copy-fusion to describe a way to break a system of mutually-recursive fixed points into nested single-variable fixed points. Like the square rule, we also need the assumption that the poset has all binary meets. Similar considerations apply.

5 Limitations

In the previous sections, we demonstrated that our graphical notation and axioms can state and prove the main theorems of the fixed-point calculus. However, there are limitations to our approach. Some logical statements cannot be translated to our graphical notation, and some problem statements are incompatible with a purely equational proof strategy and must use inequational reasoning.

5.1 Some Statements We Cannot Represent Using Diagrams

Our graphical diagrams can represent conjunctions, inequalities, monotone function application, and existential variables. We can also use equality (=) or inequality (⊆) of diagrams to represent logical equivalence and implication, as long as those appear at the very top of the formula.

We don't have good ways to represent other logical connectives, or nested implications. One example of a problematic formula is the specification for a greatest lower bound [Meertens 1998]. The greatest lower bound of a set, $\inf X$, is a lower bound that is larger than any other lower bound. (It is not the same as the least element because the greatest lower bound may not belong to the set.)

Specification 5.1. A **greatest lower bound** of a set X is an element $\inf X$ which obeys:

$$\forall x (x \in X \implies \inf X \leq x) \tag{10}$$

$$\forall x (x \in X \implies y \leq x) \implies y \leq \inf X \tag{11}$$

The above specification can be reformulated as a single equivalence. The left-to-right implication is equivalent to (10) and the right-to-left one is equivalent to (11).

Specification 5.2. A **greatest lower bound** of a set X is an element $\inf X$ which obeys:

$$y \leq \inf X \iff \forall x (x \in X \implies y \leq x)$$

Specifications 5.1 and 5.2 both feature a nested universal quantifier and a nested implication, which cannot be directly represented by our diagrams.

5.2 Asymmetrical Existence Conditions Are Incompatible With Symmetrical Proofs

Equational proofs that consist of a single chain of equations can sometimes yield multiple theorems. For example, when we set out to prove the simpler form of the rolling rule (Theorem 2.19), our calculation revealed Theorems 2.20, and 4.5. Non-equational proofs easily miss these generalizations. For instance, Backhouse [2002] proves Theorem 2.19 by mutual inclusion in a way that cannot generalize to Theorem 2.20.

The two-for-one effect is even more pronounced in theorems like the diagonal rule (Theorem 4.9). It equates two fixed points, and asks us to show that if the first fixed point exists then so does the second and vice versa. We call this a problem with a *symmetrical* existence condition. Here, equational proofs can prove both implications in one go, cutting the work in half!

However, some problems have *asymmetrical* existence conditions, in which case the symmetrical nature of our proofs comes to our detriment. An example is the stronger form of the square rule (4.24) that drops the assumption that the poset have all binary meets [Backhouse 2021]. In that version of the square rule it is still true that the existence of $\mu(f \circ f)$ implies the existence of μf, but the reverse no longer holds — for a counter-example, refer to Appendix B. Our equational proofs are incompatible with asymmetric conditions, as they cannot prove one of the existence implications without simultaneously proving its reverse. We have two choices: either we introduce assumptions to make the conditions symmetrical, or we fall back to the non-equational specification of least pre-fixed points.

The silver lining is that these subtle issues of asymmetry do not always matter in practice. It is common to reason about fixed points in settings where all conditions are symmetrical, because the functions of interest are guaranteed to have a least pre-fixed point. This is the case of monotonic functions over complete lattices (by the Knaster-Tarski Theorem) and of computable functions in the setting of denotational semantics (by Kleene's Fixed-Point Theorem).

5.3 Completeness

Last, but not least, there is the question of whether the graphical axioms presented in Section 3.3 are complete. It turns out that it is hard to pin down what this would mean in the context of the fixed-point calculus, because it is not a formal logical calculus; it is only a collection of useful theorems. We therefore do not introduce a distinction between syntax and semantics, which would usually be present when discussing string diagrams. We use string diagrams as a tool, a notation to make proofs more readable, with fewer nested quantifiers, which as we have argued also makes the proofs more discoverable.

As discussed in the previous sections, there are some logical statements that we do not know how to represent as diagrams, and for some proofs we have had to resort to inequational axioms. We currently do not have a precise characterization of what our axioms can prove.

6 Related Work

Feasibility relations have appeared in the literature under several other names: boolean-enriched profunctors [Fong and Spivak 2019], relational profunctors [Hyland and Schalk 2003], and weakening relations [Moshier 2015].

We borrow our graphical notation and most of the diagram-rewriting axioms from Piedeleu and Zanasi [2023b], which is our most direct inspiration. They applied string diagrams and feasibility relations (which they call Boolean-enriched profunctors) to develop an equational theory of finite automata equivalence. Their domain was that of regular languages, which are a particular type of partially ordered set. Their fixed points are Kleene stars, ours are fixed points of arbitrary monotonic functions. Accordingly, their theory included additional axioms that are true for regular languages but not for partially ordered sets in general. The key contribution of their work was a certain completeness theorem: they proved that their reasoning axioms can implement an automata minimation procedure and therefore can decide the equivalence of any two finite automata.

Another equational treatment of fixed points was given by Meertens [1998]. His work, which employs non-graphical point-free notation, presents equivalences for indirect equality and inequality, infima, and suprema. However, he still specifies least elements via inequalities. For this reason, he favored working with the infimum of the set of fixed points instead of the least element of the set of fixed points. The infimum cannot be used to prove that a least fixed point exists, but is equivalent if it does exist. Their specification was similar to Specification 5.2, albeit with point-free notation.

As catalogued by Backhouse, the fixed-point fusion rule is known by several names and statements. The theorem perhaps first appeared in the setting of abstract interpretation [Cousot and Cousot 1979], where it is now known as *fixpoint transfer*. Apt and Plotkin [1986] call it the *transfer lemma* and pose it with a slightly weaker hypothesis, only requiring ℓ to be strict and continuous. Gardiner and Pandya [1992] call it the *lifting lemma* and use it to reason about recursive programs.

The mutual recursion theorem was also studied by Bekić, who posed the problem in terms of directed complete partial orders (dcpos) and continuous functions [Bekić 1984]. In that context, all the requisite least-fixed points are guaranteed to exist. Technically, Bekić's theorem does not require that the poset have all binary meets, however that condition does hold in several common applications of fixed points.

7 Conclusion

In this paper we described a correct toolbox of techniques to reason equationally about partially ordered sets and least fixed points, although we do not have formal results as to its completeness.

We demonstrated the use of relational equivalences to characterize partially ordered sets, order-preserving functions and relations, least elements, and least pre-fixed and fixed points. We also presented a point-free graphical notation that is well suited for representing such relations.

We applied our techniques to construct equational proofs for key results from the fixed-point calculus of Aarts et al. [1995], including the rolling, diagonal, and square rules, fixed point fusion, and mutual recursion. With the exception of fixed-point fusion, our proofs are entirely equational.

Acknowledgments

Funded by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Project Code 403601/2023-1. We also thank the peer reviewers and the many colleagues who provided invaluable feedback to the draft of this paper.

A Non-Equational Proofs of the Rolling Rule

As we discuss in Section 2.3, there are actually two variants of the rolling rule. In both cases, if we follow the common specification of least pre-fixed points (1.1), we are encouraged to write non-equational proofs that are split into more than one part.

The simplest variant of the rolling rule was presented in the Introduction and in Theorem 2.19. A proof of it may be found in the appendix of Backhouse [2002]. It is a six-step, two-part affair that separately proves $\mu(f \circ g) \leq f(\mu(g \circ f))$ and $\mu(f \circ g) \geq f(\mu(g \circ f))$.

The stronger variant of the rolling rule is the one presented in Theorem 2.20. A proof of it may be found in Backhouse [2021]. We reproduce below an elementary version that does not employ auxiliary lemmas. Like the other proof, it is split into two parts, but this time they correspond to the two parts of Specification 1.1.

Theorem A.1. *Let $f : B \to A$ and $g : A \to B$ be monotonic functions. If $g \circ f$ has a least pre-fixed point $\mu(g \circ f)$, then $f(\mu(g \circ f))$ is a least pre-fixed point of $f \circ g$.*

Proof.
First we show that $f(\mu(g \circ f))$ is a pre-fixed point of $f \circ g$.

$$
\begin{array}{ll}
 & \text{True} \\
\Longrightarrow & \quad \{\ \mu(g \circ f) \text{ is a pre-fixed point }\} \\
 & g(f(\mu(g \circ f))) \leq \mu(g \circ f) \\
\Longrightarrow & \quad \{\ f \text{ is monotonic }\} \\
 & f(g(f(\mu(g \circ f)))) \leq f(\mu(g \circ f)
\end{array}
$$

Second, we show that it is less than or equal to all other pre-fixed points. In other words, for all x, $f(g(x)) \leq x \implies f(\mu(g \circ f)) \leq x$.

$$
\begin{array}{ll}
 & f(g(x)) \leq x \\
\Longrightarrow & \quad \{\ g \text{ is monotonic }\} \\
 & g(f(g(x))) \leq g(x) \\
\Longrightarrow & \quad \{\ \mu(g \circ f) \text{ is the least pre-fixed point }\} \\
 & \mu(g \circ f) \leq g(x) \\
\Longrightarrow & \quad \{\ f \text{ is monotonic }\} \\
 & f(\mu(g \circ f)) \leq f(g(x)) \\
\Longrightarrow & \quad \{\ \text{transitivity with } f(g(x)) \leq x\ \} \\
 & f(\mu(g \circ f)) \leq x
\end{array}
$$
□

B A Counterexample Involving the Square Rule

Let P be a poset and $f : P \to P$ be a monotonic function. The square rule (Section 4.6) states that if both f and $f \circ f$ have a least fixed point, then $\mu f = \mu(f \circ f)$. However, the existence conditions between these least fixed points are not symmetric. Although it is always true that if $f \circ f$ has a least fixed point then f also does have one, in general the reverse may not hold.

Theorem B.1. *If $f \circ f$ has a least fixed point, then f also does and $\mu f = \mu(f \circ f)$.*

Proof. First, we must show that $\mu(f \circ f)$ is a pre-fixed point of f. This follows from the rolling rule, which asserts that $f(\mu(f \circ f)) = \mu(f \circ f)$. Second, we must show that $\mu(f \circ f)$ is the less than any pre-fixed point of f. This is true because all pre-fixed points of f are also pre-fixed points of $f \circ f$: since f is monotonic, $f(x) \leq x$ implies $f(f(x)) \leq x$. Note that we were forced to use non-equational reasoning due to the assymetric nature of the theorem statement. □

Theorem B.2. *If f has a least pre-fixed point, $f \circ f$ may not have one.*

Proof. Consider a poset with three incomparable elements, A, B, C, where each element can only be compared against itself, as in $A \leq A$. Let f be the function that maps A to itself and B and C to each other. It is trivially monotonic, as all functions in this poset are.

$$\begin{tikzcd} A \arrow[loop above, "f"] & B \arrow[r, bend left, "f"] & C \arrow[l, bend left, "f"] \end{tikzcd} \qquad \begin{tikzcd} A \arrow[loop above, "f \circ f"] & B \arrow[loop above, "f \circ f"] & C \arrow[loop above, "f \circ f"] \end{tikzcd}$$

We can verify that f has a least pre-fixed point because A is its only pre-fixed point. On the contrary, $f \circ f$ has no *least* pre-fixed point; all of A, B ,C are pre-fixed points but they are incomparable. □

In widely circulated but unpublished work, Backhouse crafted conditions under which the existence of a least fixed point of f guarantees the existence of a least fixed point of $f \circ f$ [Backhouse 2021]. Because our equational techniques cannot handle asymmetric existence conditions, our version of the square rule (Thm. 4.24) requires Backhouse's hypothesis in both directions and is therefore slightly weaker.

C Proofs from Section 2

Here we provide complete proofs for all the lemmas and theorems from Section 2.

C.1 Preorders

The following lemmas culminate in the proof of Theorem 2.9.

Lemma C.1. *A relation is **transitive** if and only if*

$$x \leq y \implies \forall z(y \leq z \implies x \leq z)\,.$$

Proof. Because $x \leq y$ doesn't depend on z, we can move the $\forall z$.

$$\begin{array}{ll} & \forall x\, y\, z \left(x \leq y \implies y \leq z \implies x \leq z \right) \\ \iff & \forall x\, y \quad \left(x \leq y \implies \forall z (y \leq z \implies x \leq z) \right) \end{array}$$

□

Lemma C.2. *A relation is **reflexive** if and only if*

$$x \leq y \impliedby \forall z(y \leq z \implies x \leq z)\,.$$

Proof. We want to show that

$$\forall x (x \leq x) \tag{12a}$$

$$\iff \forall x\, y \left(\forall z (y \leq z \implies x \leq z) \implies x \leq y \right) \tag{12b}$$

- (12a) implies (12b):

$$
\begin{array}{ll}
 & \forall z (y \le z \implies x \le z) \\
\implies & \quad \{ \text{ instantiate } z := y \ \} \\
 & y \le y \implies x \le y \\
\implies & \quad \{ \text{ assumption (12a) } \} \\
 & x \le y
\end{array}
$$

- (12b) implies (12a):

$$
\begin{array}{ll}
 & \text{True} \\
\implies & \quad \{ \text{ predicate calculus } \} \\
 & \forall z (x \le z \implies x \le z) \\
\implies & \quad \{ \text{ assumption (12b) } \} \\
 & x \le x
\end{array}
$$

□

Theorem 2.9. *A relation $\le$ is a **preorder** if and only if*

$$x \le y \iff \forall z (y \le z \implies x \le z) \,.$$

Proof. Follows from Lemmas C.1 and C.2. □

C.2 Partial Orders

The following lemmas culminate in the proof of Theorem 2.10.

Lemma C.3. *A preorder is **antisymmetric** if and only if*

$$x = y \impliedby \forall z (x \le z \iff y \le z) \,.$$

Proof. We want to show that, for all x and y,

$$
\begin{array}{ll}
 & (y \le x \land x \le y) \implies x = y \\
\iff & \\
 & \forall z (x \le z \iff y \le z) \implies x = y
\end{array}
$$

We can use indirect inequality to prove that the two hypotheses are equivalent.

$$
\begin{array}{ll}
 & (y \le x \land x \le y) \\
\iff & \\
 & \forall z (x \le z \implies y \le z) \ \land \ \forall z (y \le z \implies x \le z) \\
\iff & \\
 & \forall z (x \le z \implies y \le z \ \land \ y \le z \implies x \le z) \\
\iff & \\
 & \forall z (x \le z \iff y \le z)
\end{array}
$$

□

Lemma C.4. *It is always true that*

$$x = y \implies \forall z (x \le z \iff y \le z)$$

Proof. Substitution of equals by equals. □

Theorem 2.10. *A preorder is **antisymmetric** if and only if*

$$x = y \iff \forall z (x \le z \iff y \le z) \,.$$

Proof. Follows from C.3 and C.4. □

C.3 Monotonicity

The following lemmas culminate in the proof of Theorem 2.11.

Lemma C.5. *A function f between preordered sets is **monotonic** iff*

$$\exists y (x \le y \land f\, y \le z) \implies f\, x \le z \,.$$

Proof. Split $f\,x \le f\,y$ via indirect inequality and move the quantifiers around.

$$
\begin{array}{ll}
 & \forall x\, y \quad (x \le y \implies f\, x \le f\, y) \\
\iff & \\
 & \forall x\, y \quad (x \le y \implies \forall z (f\, y \le z \implies f\, x \le z)) \\
\iff &
\end{array}
$$

$$
\begin{array}{ll}
 & \forall x\, y\, z\, \big(\, x \le y \implies f\, y \le z \implies f\, x \le z \,\big) \\
\iff & \forall x\, y\, z\, \big(\, (x \le y \land f\, y \le z) \implies f\, x \le z \,\big) \\
\iff & \forall x\, z \quad \big(\, \exists y (x \le y \land f\, y \le z) \implies f\, x \le z \,\big)
\end{array}
$$
□

LEMMA C.6. *It is always true that*

$$\exists y (x \le y \land f\, y \le z) \impliedby f\, x \le z\,.$$

PROOF. Satisfy the existential with $y := x$. □

THEOREM 2.11. *A function f between preordered sets is **monotonic** iff*

$$\exists y (x \le y \land f\, y \le z) \iff f\, x \le z\,.$$

PROOF. Follows from C.5 and C.6. □

C.4 Corollaries of Indirect Monotonicity

PROOF OF COROLLARY 2.12. Theorem 2.11 with $f(x) = x$. □

PROOF OF COROLLARY 2.13. Theorem 2.11 with $f(x) = (x, x)$. □

C.5 Least Element

The standard definition of the least element asks that the least element should belong to the set and that it be a lower bound. It turns out that it will be convenient to work with a slightly weaker membership condition.

The following three lemmas will be used in the proof of Theorem 2.14.

LEMMA C.7. *m is the **least element** of the poset X if and only if it obeys*

$$\exists x (x \in X \land x \le m) \qquad \text{Weak membership}$$

$$\forall x (x \in X \implies m \le x) \qquad \text{Lower bound}$$

PROOF. It suffices to show that the two membership conditions are equivalent under the shared requirement that the least element is a lower bound. That is, we assume

$$\forall x (x \in X \implies m \le x) \tag{13}$$

and we want to show that

$$
\begin{array}{lll}
 & m \in X & \text{(14a)} \\
\iff & \exists x (x \in X \land x \le m)\,. & \text{(14b)}
\end{array}
$$

- (14a) implies (14b):

$$
\begin{array}{ll}
 & m \in X \\
\implies & m \in X \land m \le m \\
\implies & \exists x (x \in X \land x \le m)
\end{array}
$$

- (14a) is implied by (14b) and (13):

$$
\begin{array}{ll}
 & x \in X \land x \le m \\
\implies & \quad \{\ \text{assumption (13)}\ \} \\
 & x \in X \land x \le m \land m \le x \\
\implies & \quad \{\ \text{by antisymmetry, } x = m\ \} \\
 & m \in X
\end{array}
$$
□

LEMMA C.8. *In a preordered set, the **weak membership** condition is equivalent to*

$$\forall y \big(\, m \le y \implies \exists x (x \in X \land x \le y) \,\big)\,.$$

Proof. We want to show that

$$\begin{aligned} & \exists x(x \in X \wedge x \leq m) && \text{(15a)} \\ \iff \; & \forall y\big(m \leq y \implies \exists x(x \in X \wedge x \leq y)\big) && \text{(15b)} \end{aligned}$$

- (15a) implies (15b):

$$\begin{aligned} & m \leq y \\ \implies & \quad \{ \text{ assumption (15a) } \} \\ & m \leq y \wedge \exists x(x \in X \wedge x \leq m) \\ \implies & \quad \{ \text{ transitivity } \} \\ & \exists x(x \in X \wedge x \leq y) \end{aligned}$$

- (15b) implies (15a):

$$\begin{aligned} & \text{True} \\ \implies & \quad \{ \text{ reflexivity } \} \\ & m \leq m \\ \implies & \quad \{ \text{ assumption (15b) with } y := m \} \\ & \exists x(x \in X \wedge x \leq m) \end{aligned}$$

□

Lemma C.9. *In a preordered set, the* ***lower bound*** *condition is equivalent to*

$$\forall y\big(m \leq y \impliedby \exists x(x \in X \wedge x \leq y)\big)$$

Proof. Rewrite $m \leq x$ with indirect inequality and then move quantifiers around.

$$\begin{aligned} & \forall x \quad \big(x \in X \implies m \leq x\big) \\ \iff & \forall x \quad \big(x \in X \implies \forall y(x \leq y \implies m \leq y)\big) \\ \iff & \forall x\, y\, \big(x \in X \implies x \leq y \implies m \leq y\big) \\ \iff & \forall x\, y\, \big((x \in X \wedge x \leq y) \implies m \leq y\big) \\ \iff & \forall y \quad \big(\exists x(x \in X \wedge x \leq y) \implies m \leq y\big) \end{aligned}$$

□

Theorem 2.14. *m is the* ***least element*** *of the poset X if and only if, for all y,*

$$m \leq y \iff \exists x(x \in X \wedge x \leq y)\,.$$

Proof. Follows from C.7 with C.8 and C.9. □

D Proofs from Section 4.3

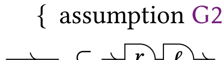
The graphical cancellation rules of Galois Connections (G5–G6) can be derived from (G2). The pointful cancellation rules (G3–G4) can in turn be derived via indirect inequality.

Proof of G2 ⇒ G5.

True

$\implies$ { rule C6 }

$\implies$ { assumption G2 }

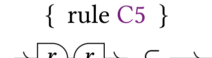


□

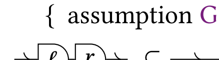
Proof of G2 ⇒ G6.

True

$\implies$ { rule C5 }

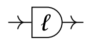


$\implies$ { assumption G2 }

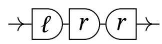


□

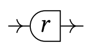
Proof of G5 ∧ G6 ⇒ G2. We show that by mutual inclusion:

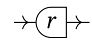


$\subseteq$ { rule C6 }



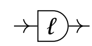
$\subseteq$ { assumption G6 }





$\subseteq$ { assumption G5 }

$\subseteq$ { rule C5 }



□

E Explanation of graphical axioms

Here we translate the graphical axioms from the main paper into pointful notation and prove that they are correct. The axioms are reproduced in Figure 3. To streamline the pointful presentation, we write relational equivalences $R = S$ in terms of mutual inclusions like $(x, y) \in R \iff (x, y) \in S$. For instance, we may write

$$x \leq y \iff \exists a(x \leq a \leq y)$$

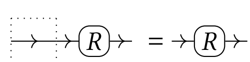
instead of writing in the set-comprehension style like

$$\{\, x, y \mid x \leq y \,\} = \{\, x, y \mid \exists a(x \leq a \leq y) \,\}\,.$$

E.1 The A block

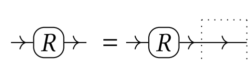
These structural rules allow us to manipulate the nesting and ordering of subrelations.

Proof of A1.



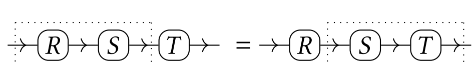
$$\exists a(x \leq a \wedge a\ R\ y) \iff x\ R\ y$$

Proof of A2.



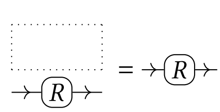
$$x\ R\ y \iff \exists b(x\ R\ b \wedge b \leq y)$$

Proof of A3.



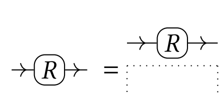
$$\exists b(\exists a(x\ R\ a\ S\ b) \wedge b\ T\ y) \iff \exists a \exists b(x\ R\ a\ S\ b\ T\ y) \iff \exists a(x\ R\ a \wedge \exists b(a\ S\ b\ T\ y))$$

Proof of A4.



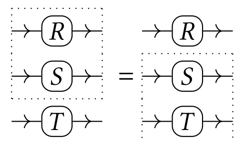
$$(\text{True} \wedge x\ R\ y) \iff x\ R\ y$$

Proof of A5.



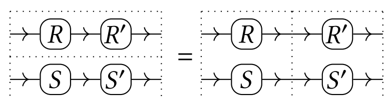
$$x\ R\ y \iff (x\ R\ y \wedge \text{True})$$

Proof of A6.



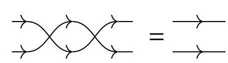
$$(x_1\ R\ y_1 \wedge x_2\ S\ y_2) \wedge x_3\ T\ y_3 \iff x_1\ R\ y_1 \wedge (x_2\ S\ y_2 \wedge x_3\ T\ y_3)$$

Proof of A7.



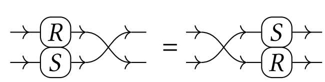
$$\begin{matrix}\exists a_1(x_1\ R\ a_1\ R'\ y_1)\\ \exists a_2(x_2\ S\ a_2\ S'\ y_2)\end{matrix} \iff \exists a_1\exists a_2\left(\begin{pmatrix}x_1\ R\ a_1\\ x_2\ S\ a_2\end{pmatrix}\wedge\begin{pmatrix}a_1\ R'\ y_1\\ a_2\ S'\ y_2\end{pmatrix}\right)$$

Proof of A8.



$$\exists a_1\exists a_2\begin{pmatrix}x_1 \leq a_2 \leq y_1\\ x_2 \leq a_1 \leq y_2\end{pmatrix} \iff \begin{matrix}x_1 \leq y_1\\ x_2 \leq y_2\end{matrix}$$

Proof of A9.

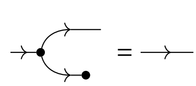


$$\exists b_1\exists b_2\begin{pmatrix}x_1\ R\ b_1 & b_1 \leq y_2\\ x_2\ S\ b_2 & b_2 \leq y_1\end{pmatrix} \iff \begin{matrix}x_1\ R\ y_2\\ x_2\ S\ y_1\end{matrix} \iff \exists a_1\exists a_2\begin{pmatrix}x_1 \leq a_1 & a_1\ R\ y_2\\ x_2 \leq a_2 & a_2\ S\ y_1\end{pmatrix}$$

E.2 The B block

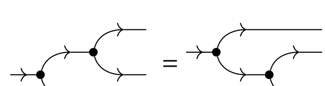
Rules B1–B6 boil down to applications of reflexivity and indirect monotonicity. Rules B7–B10 are only inclusions ($\subseteq$), because a suitable a or b might not exist.

Proof of B1.



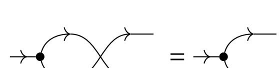
$$\exists a\begin{pmatrix}x \leq y\\ x \leq a\end{pmatrix} \iff x \leq y$$

Proof of B2.



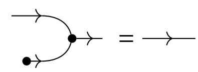
$$\exists a\begin{pmatrix}x \leq a & a \leq y_1\\ x \leq y_3 & a \leq y_2\end{pmatrix} \iff \begin{matrix}x \leq y_1\\ x \leq y_2\\ x \leq y_3\end{matrix} \iff \exists b\begin{pmatrix}x \leq y_1 & b \leq y_3\\ x \leq b & b \leq y_3\end{pmatrix}$$

Proof of B3.



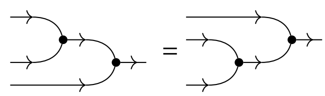
$$\exists a_1 \exists a_2 \begin{pmatrix} x \le a_2 \le y_1 \\ x \le a_1 \le y_2 \end{pmatrix} \iff \begin{array}{l} x \le y_1 \\ x \le y_2 \end{array}$$

Proof of B4.



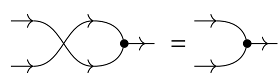
$$\exists a \begin{pmatrix} x \le y \\ a \le y \end{pmatrix} \iff x \le y$$

Proof of B5.



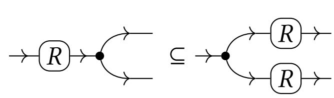
$$\exists a \left(\begin{array}{ll} x_1 \le a & a \le y \\ x_2 \le a & x_3 \le y \end{array} \right) \iff \begin{array}{l} x_1 \le y \\ x_2 \le y \\ x_3 \le y \end{array} \iff \exists a \left(\begin{array}{ll} x_2 \le a & x_1 \le y \\ x_2 \le a & a \le y \end{array} \right)$$

Proof of B6.



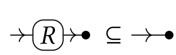
$$\exists a \begin{pmatrix} x_1 \le a_2 \le y \\ x_2 \le a_1 \le y \end{pmatrix} \iff \begin{array}{l} x_1 \le y \\ x_2 \le y \end{array}$$

Proof of B7.



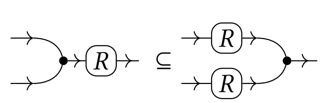
$$\exists b \left(\begin{array}{ll} x \; R \; b & \begin{array}{l} b \le y_1 \\ b \le y_2 \end{array} \end{array} \right) \implies \begin{array}{l} x \; R \; y_1 \\ x \; R \; y_2 \end{array} \iff \exists a_1 \exists a_2 \left(\begin{array}{ll} x \le a_1 & a_1 \; R \; y_1 \\ x \le a_2 & a_2 \; R \; y_2 \end{array} \right)$$

Proof of B8.



$$\exists b (x \; R \; b) \implies \text{True}$$

Proof of B9.



$$\exists a \left(\begin{array}{ll} \begin{array}{l} x_1 \le a \\ x_2 \le a \end{array} & a \; R \; y \end{array} \right) \implies \begin{array}{l} x_1 \; R \; y \\ x_2 \; R \; y \end{array} \iff \exists b_1 \exists b_2 \left(\begin{array}{ll} x_1 \; R \; b_1 & b_1 \le y \\ x_2 \; R \; b_2 & b_2 \le y \end{array} \right)$$

Proof of B10.

$$\exists a (a \; R \; y) \implies \text{True}$$

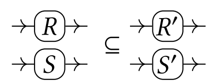
Proof of B11.

$$\begin{array}{l} R \subseteq R' \\ S \subseteq S' \end{array} \implies$$

$$\begin{pmatrix} \forall x \forall y (x \; R \; y \implies x \; R' \; y) \\ \forall y \forall z (y \; S \; z \implies y \; S' \; z) \end{pmatrix} \implies \big(\exists y (x \; R \; y \; S \; z) \implies \exists y (x \; R' \; y \; S' \; z)\big)$$

Proof of B12.

$$\begin{matrix} R \subseteq R' \\ S \subseteq S' \end{matrix} \implies$$

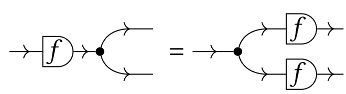


$$\begin{pmatrix} \forall x_1 \, y_1 \, (x_1 \; R \; y_1 \implies x_1 \; R' \; y_1) \\ \forall x_2 \, y_1 \, (x_2 \; S \; y_2 \implies x_2 \; S' \; y_2) \end{pmatrix} \implies \begin{pmatrix} x_1 \; R \; y_1 & \implies & x_1 \; R' \; y_1 \\ x_2 \; S \; y_2 & \implies & x_2 \; S' \; y_2 \end{pmatrix}$$

E.3 The C block

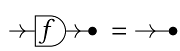
Rules C1–C4 show that several relational inclusions become equalities when working with functions. Rule C5 and C6 respectively boil down to transitivity and monotonicity.

Proof of C1.



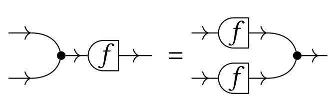
$$\exists b \begin{pmatrix} f\,x \le b & \begin{matrix} b \le y_1 \\ b \le y_2 \end{matrix} \end{pmatrix} \iff \begin{matrix} f\,x \le y_1 \\ f\,x \le y_2 \end{matrix} \iff \exists a_1 \exists a_2 \begin{pmatrix} x \le a_1 & f\,a_1 \le y_1 \\ x \le a_2 & f\,a_2 \le y_2 \end{pmatrix}$$

Proof of C2.



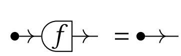
$$\exists b (f\,x \le b) \iff \text{True}$$

Proof of C3.



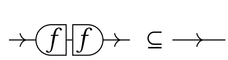
$$\exists a \begin{pmatrix} \begin{matrix} x_1 \le a \\ x_2 \le a \end{matrix} & a \le f\,y \end{pmatrix} \iff \begin{matrix} x_1 \le f\,y \\ x_2 \le f\,y \end{matrix} \iff \exists b_1 \exists b_2 \begin{pmatrix} x_1 \le f\,b_1 & b_1 \le y \\ x_2 \le f\,b_2 & b_2 \le y \end{pmatrix}$$

Proof of C4.



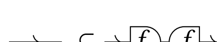
$$\exists a (a \le f\,y) \iff \text{True}$$

Proof of C5.



$$\exists a (x \le f\,a \le y) \implies x \le y$$

Proof of C6.

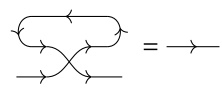


$$x \le y \implies \exists b (f\,x \le b \le f\,y)$$

E.4 The D block

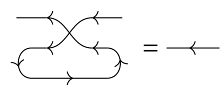
These rules describe properties of the looping morphisms. To keep things short, we only create one existential variable per wire, instead of one at every single connection point between generators.

Proof of D1.



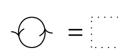
$$\exists a(x \leq a \leq y) \iff x \leq y$$

Proof of D2.



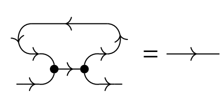
$$\exists a(x \geq a \geq y) \iff x \geq y$$

Proof of D3.



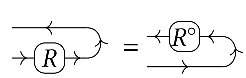
$$\exists a \exists b \begin{pmatrix} a \leq b \\ b \leq a \end{pmatrix} \iff \text{True}$$

Proof of D4.



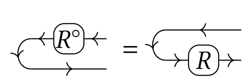
$$\exists ab \begin{pmatrix} b \leq a \leq b \\ x \leq a \leq y \end{pmatrix} \iff x \leq y$$

Proof of D5.



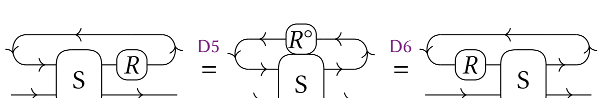
$$\exists b \begin{pmatrix} y \geq b \\ x \; R \; b \end{pmatrix} \iff x \; R \; y \iff \exists a \begin{pmatrix} y \; R^{\circ} \; a \\ x \leq a \end{pmatrix}$$

Proof of D6.



$$\exists b \begin{pmatrix} b \; R^{\circ} \; x \\ b \leq y \end{pmatrix} \iff x \; R \; y \iff \exists a \begin{pmatrix} a \geq x \\ a \; R \; y \end{pmatrix}$$

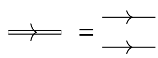
Proof of D7.



E.5 The E block

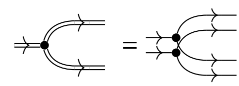
Rules E1–E8 state that wires transporting a product poset $P \times Q$ may be split into parallel wires. This boils down to a comparison at the level of the product poset being equivalent to element-wise comparison of the two components. We show only the first two rules, as the rest are similar.

Proof of E1.



$$((x_1, x_2) \leq (y_1, y_2)) \iff \begin{matrix} x_1 \leq y_1 \\ x_2 \leq y_2 \end{matrix}$$

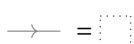
Proof of E2.



$$\begin{pmatrix} (x_1, x_2) \leq (y_1, y_2) \\ (x_1, x_2) \leq (z_1, z_2) \end{pmatrix} \iff \begin{pmatrix} x_1 \leq y_1 & x_2 \leq y_2 \\ x_1 \leq z_1 & x_2 \leq z_2 \end{pmatrix}$$

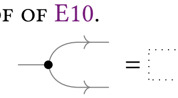
Rules E9–E16 govern wires in the unitary poset. Since there is a single element, comparisons are always true due to reflexivity. Again, we show only the fist two rules.

Proof of E9.



$$(\bullet \leq \bullet) \iff \text{True}$$

Proof of E10.



$$\begin{pmatrix} \bullet \leq \bullet \\ \bullet \leq \bullet \end{pmatrix} \iff \text{True}$$

References

Chritiene Aarts, Roland Backhouse, Eerke Boiten, Henk Doornbos, Netty van Gasteren, Rik van Geldrop, Paul Hoogendijk, Ed Voermans, and Jaap van der Woude. 1995. Fixed-point calculus. *Inform. Process. Lett.* 53, 3 (1995), 131–136. doi:10.1016/0020-0190(94)00195-5

K. R. Apt and G. D. Plotkin. 1986. Countable nondeterminism and random assignment. *J. ACM* 33, 4 (Aug. 1986), 724–767. doi:10.1145/6490.6494

Ralph-Johan Back and Joakim von Wright. 1999. Reasoning algebraically about loops. *Acta Informatica* 36, 4 (1999), 295–334. doi:10.1007/s002360050163

Ralph-Johan Back and Joakim von Wright. 2012. *Refinement calculus: a systematic introduction.* Springer Science & Business Media, New York, NY, USA. doi:10.1007/978-1-4612-1674-2

Roland Backhouse. 2002. Galois Connections and Fixed Point Calculus. In *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction: International Summer School and Workshop Oxford, UK, April 10–14, 2000 Revised Lectures.* Springer, Berlin, Heidelberg, 89–150. doi:10.1007/3-540-47797-7_4

Roland Backhouse. 2021. Mathematics of Program Construction. Unpublished draft book.

Hans Bekić. 1984. Definable Operation in General Algebras, and the Theory of Automata and Flowcharts. In *Programming Languages and Their Definition*, G. Goos and J. Hartmanis (Eds.). Springer-Verlag, Berlin, Heidelberg, 30–55. doi:10.1007/BFb0048939

Carsten Butz. 1998. *Regular Categories and Regular Logic.* Technical Report. University of Aarhus. BRICS Lecture Series LS-98-2.

Andrea Censi. 2015. A Mathematical Theory of Co-Design. *CoRR* abs/1512.08055 (2015), 18 pages. arXiv:1512.08055 doi:10.48550/arXiv.1512.08055

Patrick Cousot and Radhia Cousot. 1979. Systematic design of program analysis frameworks. In *Proceedings of the 6th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages* (San Antonio, Texas) *(POPL '79).* Association for Computing Machinery, New York, NY, USA, 269–282. doi:10.1145/567752.567778 Note: Fixed-point fusion is Theorem 7.1.0.4.

Jules Desharnais, Bernhard Möller, and Georg Struth. 2011. Algebraic Notions of Termination. *Logical Methods in Computer Science* 7, 1 (2011), 1–29. doi:10.2168/LMCS-7(1:1)2011

Edsger W. Dijkstra. 1991. Why preorders are beautiful. (June 1991). http://www.cs.utexas.edu/users/EWD/ewd11xx/EWD1102.PDF EWD 1102; circulated privately.

Brendan Fong and David I. Spivak. 2019. *An Invitation to Applied Category Theory: Seven sketches in compositionality.* Cambridge University Press, Cambridge, UK. doi:10.1017/9781108668804

P. H. B. Gardiner and P. K. Pandya. 1992. Reasoning algebraically about recursion. *Sci. Comput. Program.* 18, 3 (June 1992), 271–280. doi:10.1016/0167-6423(92)90019-8

Andy Gill and Graham Hutton. 2009. The worker/wrapper transformation. *Journal of Functional Programming* 19, 2 (2009), 227–251. doi:10.1017/S0956796809007175

Martin Hyland and Andrea Schalk. 2003. Glueing and orthogonality for models of linear logic. *Theoretical computer science* 294, 1-2 (2003), 183–231. doi:10.1016/S0304-3975(01)00241-9

Lambert Meertens. 1998. *Least Fixpoints Calculationally.* Technical Report. Kestrel Institute.

Andrew M Moshier. 2015. Coherence for categories of posets with applications. In *Topology, Algebra and Categories in Logic (TACL).*

Shin-Cheng Mu, Hsiang-Shang Ko, and Patrik Jansson. 2009. Algebra of programming in Agda: dependent types for relational program derivation. *Journal of Functional Programming* 19, 5 (2009), 545–579. doi:10.1017/S0956796809007345

Robin Piedeleu and Fabio Zanasi. 2023a. A Finite Axiomatisation of Finite-State Automata Using String Diagrams. *Logical Methods in Computer Science* 19, 1 (2023), 13–38. doi:10.46298/lmcs-19(1:13)2023

Robin Piedeleu and Fabio Zanasi. 2023b. An Introduction to String Diagrams for Computer Scientists. (2023). arXiv:2305.08768 [cs.LO] doi:10.48550/arXiv.2305.08768

Neil Sculthorpe and Graham Hutton. 2014. Work it, wrap it, fix it, fold it. *Journal of Functional Programming* 24, 1 (2014), 113–127. doi:10.1017/S0956796814000045

Peter Selinger. 2011. *A Survey of Graphical Languages for Monoidal Categories.* Springer, Berlin, Heidelberg, 289–355. doi:10.1007/978-3-642-12821-9_4

Received 2026-02-19; accepted 2026-05-13