UNIVERSITY OF CAMBRIDGE
Department of Computer Science and Technology

# Univalent Semantics for Varying Evaluation Strategies

Beatrice Szilvasy

St Catharine's College

May 15, 2026

Submitted in partial fulfillment of the requirements for the Computer Science Tripos, Part III

Total page count: 62

Main chapters (excluding front-matter, references and appendix): 48 pages (pp 1–48)

Main chapters word count: 10658 (text: 9256, headers: 72, captions/footnotes: 257, inlinemath: 983, displaymath: 90)

Methodology used to generate that word count:

```
texcount -template="{SUM} (text: {word}, headers: {headerword}, "\
"captions/footnotes: {otherword}, inlinemath: {inline}, "\
"displaymath: {displaymath})" \
  $FILE.tex
```

With front- and backmatter ignored within `$FILE.tex` using `%TC:ignore`.

# Declaration

I, Beatrice Szilvasy of St Catharine's College, being a candidate for Part III of the Computer Science Tripos, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this report, I adhered to the Department of Computer Science and Technology AI Policy. [The project required the approved use of {insert name of technology} and such use is acknowledged in the text at the relevant sections.] [I am content for my report to be made available to the students and staff of the University.]

**Signed:**

**Date:**

# Abstract

TODO: this is the original as in the proposal. I will revise this as the work gets wrapped up.

Call-by-value and call-by-name are dual evaluation strategies in programming languages; when these are mixed in an effectful language, function composition ceases to be associative, rendering ordinary categories unsuitable for modelling their denotational semantics. Suitable constructs such as duploids have been studied in previous work, but in a univalent setting it is desirable to work with "univalent" versions, where equivalences correspond to identity types. In this work we will study and formalise the suitable notions of univalence for these constructs using the Rocq UniMath library.

# Acknowledgements

This project would not have been possible without the wonderful support of ….

# Contents

# 1. HALF-POLISHED Introduction

This dissertation is concerned with the semantics of programming languages in which composition is non-associative. That is, languages in which valid programs

$$\texttt{let } y := (\texttt{let } x := e_1 \texttt{ in } e_2) \texttt{ in } e_3 \tag{1.1}$$

and

$$\texttt{let } x := e_1 \texttt{ in } (\texttt{let } y := e_2 \texttt{ in } e_3) \tag{1.2}$$

(where $e_3$ does not reference $x$) may compute differently.[1] One way in which such programs arise is in the presence of side effects with mixed evaluation order.

To illustrate the effect of evaluation order, consider the program:

$$\texttt{let } x := (\texttt{print("hello"); } 1) \texttt{ in } x + x\,. \tag{1.3}$$

Here, evaluating $\texttt{print}(v)$ has the side-effect of outputting $v$ to a console. There are two canonical evaluations of Program 1.3 depending on the choice of evaluation order for the "`let`" construct.

- Using *call-by-name* (CBN), evaluate the expression $\texttt{let } x := e_1 \texttt{ in } e_2$ by first substituting $e_1$ for $x$ in $e_2$. Then evaluate the resulting expression $e_1[x := e_2]$. Using call-by-name, Program 1.3 outputs `hellohello` and returns 2.
- Using *call-by-value* (CBV), evaluate $\texttt{let } x := e_1 \texttt{ in } e_2$ by first evaluating $e_1$ to its value $v$. Then substitute $v$ for $x$ in $e_2$. Finally, evaluate the resulting expression $e_2[x := v]$. Using call-by-value, Program 1.3 outputs just `hello` and returns 2.

So long as all `let` expressions in a language use the same choice of evaluation order out of call-by-value or call-by-name, composition will remain associative. Combining call-by-value and call-by-name in a single language exposes non-associative composition.

[1] TODO: Perhaps Levy's "let $e_1$ be $x$ in $e_2$" syntax would be better?

Let us write

$$\texttt{let}\ x \overset{\mathrm{V}}{\coloneqq} e_1\ \texttt{in}\ e_2 \qquad \text{for call-by-value, and}$$
$$\texttt{let}\ x \overset{\mathrm{N}}{\coloneqq} e_1\ \texttt{in}\ e_2 \qquad \text{for call-by-name.}$$

Then consider, for an example of non-associativity, the program

$$\begin{aligned} &\texttt{let}\ y \overset{\mathrm{N}}{\coloneqq} (\texttt{let}\ x \overset{\mathrm{V}}{\coloneqq} (\texttt{print("hello");}\ 1) \\ &\qquad\qquad \texttt{in}\ (\texttt{print("world");}\ x))\ \texttt{in}\ y + y \end{aligned} \tag{1.4}$$

of the left-associated form (like Program 1.1), and

$$\begin{aligned} &\texttt{let}\ x \overset{\mathrm{V}}{\coloneqq} (\texttt{print("hello");}\ 1)\ \texttt{in} \\ &(\texttt{let}\ y \overset{\mathrm{N}}{\coloneqq} (\texttt{print("world");}\ x)\ \texttt{in}\ y + y) \end{aligned} \tag{1.5}$$

its reassociation to the right (like Program 1.2). Recall that call-by-name `let` substitutes the entire assigned expression into the body, while call-by-value `let` evaluates it first and then substitutes. Then we can see that evaluating Program 1.4 will output **`helloworldhelloworld`**, but Program 1.5 will output **`helloworldworld`**, and both will return 2. Thus we have two programs that differ only up to associativity, and yet compute differently.

It is typical to interpret the semantics of programming languages in *categories* (Definition 3.1), but this breaks down for modelling non-associative computation. Let us consider $\texttt{let}\ y \coloneqq (\texttt{let}\ x \coloneqq e_1\ \texttt{in}\ e_2)\ \texttt{in}\ e_3$ (Program 1.1) and $\texttt{let}\ x \coloneqq e_1\ \texttt{in}\ (\texttt{let}\ y \coloneqq e_2\ \texttt{in}\ e_3)$ (Program 1.2) to be interpreted in some category $\mathcal{C}$, and assign to each expression $x : A \vdash e : B$ some morphism $[\![e]\!] : [\![A]\!] \to [\![B]\!]$ in $\mathcal{C}$. The question of whether Program 1.1 and Program 1.2 compute differently then amounts, informally,[2] to the following putative equation:

$$\Big( [\![e_1]\!] \mathbin{;} [\![e_2]\!] \Big) \mathbin{;} [\![e_3]\!] \overset{?}{=} [\![e_1]\!] \mathbin{;} \Big( [\![e_2]\!] \mathbin{;} [\![e_3]\!] \Big). \tag{1.6}$$

However, this is trivialized by the associativity of composition (Definition 3.1.4.b) in any category $\mathcal{C}$. If we wish to take seriously the idea that equations of the form in Equation 1.6 do not necessarily hold, then we must use a different, more general model. In this dissertation, it will be Munch-Maccagnoni's *duploids* [23] that generalize and

[2]This would normally require extra bookkeeping to keep multiple variables in scope.

take the role of categories for modelling non-associative computation.

[TODO: mention monads and thunking CPS translations and their issues, to head off those familiar. Will do so when revising the background chapters.] For what purpose should we mix evaluation orders? One motivation is that it allows subsuming the theories of both call-by-name and call-by-value into a single paradigm such as *call-by-push-value* [16] (CBPV). The denotational semantics of call-by-push-value can be modelled "indirectly" by adjunctions [15], generalizing the widespread use of monads for the semantics of call-by-value [20]. Duploids fit into the picture by providing "direct" denotational semantics for mixed evaluation order, and have a correspondence with adjunctions [23].

The purpose of this dissertation is to develop the theory of duploids formally in (a variant of) Martin Löf type theory and from a *univalent point of view* [26] (see Section 2.2). [TODO: explain UF, briefly? I don't want to dwell on this too long in the introduction because it needs too much background, which is what the next chapter is for.] We[3] formalize some existing results about duploids within the Rocq library UniMath [10]. Some of this work involves transferring existing theorems over to a more recent definition of duploids (see Remark 4.4). [NOTE: would be really cool to prove the full "structure theorem" for single-sorted duploids, since that does not yet exist in the literature.] We define what it means for a duploid $\mathcal{D}$ to be *univalent*: that isomorphisms $a \cong b$ in $\mathcal{D}$ (of a suitable kind) render the related objects $a, b : \mathcal{D}$ indistinguishable to the type theory. We show that the original construction of the duploid arising from an adjunction fails to be univalent, and then provide a novel construction of the same duploid which **is** univalent.

## 1.1. NOTES Outline

**Rendered**: May 15, 2026

These are my running notes about the structure of this text. Chapters are marked (in the TOC and their heading) with their completion status:

- NOTES means (rendered) comments from me to the reader of the draft (also typically me). These are also scattered through the draft as TODO, FIXME and NOTE tags, so that the reader knows which parts are incomplete and how.

[3] Me, my notebooks, and my proverbial rubber duck.

- TODO means that the section has not been written, or at least is not yet structured in a way that is actually coherent. Paragraphs may need to be rewritten or moved to different sections.
- VERY-ROUGH and ROUGH mean that some content exists and is intended to be legible, but content may be lacking and some things might not be coherent.
- HALF-POLISHED means the section is intended to be comprehensible as-is to the target audience, although is intended to undergo further refinement.
- "DONE" may happen in the future.

# 2. ROUGH Background

## 2.1. ROUGH Mixed Evaluation Order

It is a well-established tradition in theoretical computer science to consider programs as terms in variants of the lambda calculus. Although Church originally studied a *pure*, side-effect free, lambda calculus for the purposes of logic [4], many languages of practical use include *computational effects*, or simply *effects*, such as input/output (IO), mutable state, or non-termination.

A simply typed lambda calculus term $\Gamma \vdash e : A$ may be interpreted in any cartesian closed category as a morphism $[\![e]\!] : [\![\Gamma]\!] \to [\![A]\!]$ [14]. This interpretation is sound (and complete), in that if two terms $\Gamma \vdash e_1, e_2 : A$ are $\beta\eta$-equivalent, then their interpretations $[\![e_1]\!], [\![e_2]\!] : [\![\Gamma]\!] \to [\![A]\!]$ are equal in all cartesian closed categories (and vice-versa).

Pointing out that the pure lambda calculus's $\beta\eta$-equivalence is the wrong notion of program equivalence in the presence of computational effects, Moggi [20, 21] suggested the use of monads to model the semantics of effectful call-by-value lambda calculi instead. This way, given a suitable monad

$$(\mathcal{C} \xrightarrow{T} \mathcal{C}, \quad \mathsf{Id}_{\mathcal{C}} \overset{\eta}{\Rightarrow} T, \quad T^2 \overset{\mu}{\Rightarrow} T)$$

on a category $\mathcal{C}$, a compuational lambda term $\Gamma \vdash e : A$ can be interpreted as a morphism $[\![e]\!] : [\![\Gamma]\!] \to T\,[\![A]\!]$ in $\mathcal{C}$. The objects of $a : \mathcal{C}$ are the types of **values**, while objects $Ta : \mathcal{C}$ are the types of **computations** producing values of the type $a : \mathcal{C}$.

Moggi's interpretation of the computational lambda calculus involves a sort of "compilation" of lambda terms into a monadic style, inserting monad operations—units and Kleisli extensions, often called "return" and "bind"—that were not present in the source. In the parlance of Führmann [8], monads constitute an *indirect semantics* for the call-by-value computational lambda calculus, in contrast to the *direct semantics* provided by, say, cartesian closed categories for the pure lambda calculus. The difference being that the calculus's syntax is much more closely related to the constructs readily available in a direct semantics, whereas an indirect semantics requires an extra transformation step. Führmann [8] provides a direct semantics for the computational lambda calculus which he calls *abstract Kleisli categories* but which we shall call *thunk-force categories* [23].

Dually, it is possible to use comonads to model the semantics of call-by-name lambda calculi [15, 23]. This way, given a suitable comonad $U\colon \mathcal{D} \to \mathcal{D}$ on a category $\mathcal{D}$, we can interpret effectful call-by-name lambda terms $\Gamma \vdash e : A$ as morphisms $[\![e]\!] : U\,[\![\Gamma]\!] \to [\![A]\!]$. The dualized analogue of a thunk-force category is a *runnable monad* [23]. TODO: the types $a : \mathcal{D}$ are the stacks, but what are the objects $Ua : \mathcal{D}$?

The use of a single category for the semantics of programming languages forces the choice of a fixed evaluation order. Otherwise, as seen in the introduction, composition ceases to be associative, and the use of categories becomes impossible.

This is undesirable for a number of reasons. Firstly, as argued by Levy [16], the combination of call-by-value and call-by-name into a single subsuming paradigm, call-by-push-value, obviates the need to reason about call-by-value and call-by-name separately. This reduces the burden of reasoning about models for both paradigms to reasoning about only one. Call-by-push-value has an indirect semantics in certain adjunctions [15]. Both call-by-value and call-by-name have reasonable translations into call-by-push-value in a way that preserves their equational theory and semantics.

Another motivation is advanced in Munch-Maccagnoni's thesis [24]. The relaxation of associativity captures meaningful properties about the types and programs, namely polarization. Guillaume proposes a direct semantics for mixed evaluation order called *duploids* [23]. Duploids subsume the direct semantics of thunk-force categories and runnable monads, and they also subsume the use of monads and comonads for call-by-value and call-by-name. Specifically, each adjunction $L \dashv R\colon \mathcal{P} \to \mathcal{N}$ gives rise to a duploid $\mathcal{E}[L \dashv R]$ which contains as subcategories the Kleisli category of $RL\colon \mathcal{P} \to \mathcal{P}$, and the co-Kleisli category of $LR\colon \mathcal{N} \to \mathcal{N}$.

The original study of duploids was done on what I call *two-sorted duploids*. These have two separate sorts of object: one sort of **positive** objects, or **values**; and one sort of **negative** objects, or **stacks**. In this work we shall study *single-sorted duploids* instead. In this case, positive and negative objects are distinguished not syntactically, but *semantically* by an associativity condition. [TODO: I could do well being more specific about these things.]

Although the background and context of this work lies in the semantics of programming languages, programming languages themselves will play little role beyond this section. Instead, we focus our attention entirely on the mathematical structures themselves.

## 2.2. HALF-POLISHED Preliminary Type Theory

This dissertation consists of an English language discussion about a body of mathematics that takes place in the formal language of homotopy type theory. The theorems, definitions, and proofs are presented in ordinary mathematical English, but they are also assigned meaning as types and terms in the language of homotopy type theory. In this section I explain some notation and terminology I will use throughout.

Some knowledge of homotopy type theory and univalent foundations is recommended for readers, as it is necessary to understand the motivations and main results. A sufficient introduction to univalent mathematics as a whole has been written by Grayson [9]. For textbook accounts, I suggest Rijke's *Introduction to Homotopy Type Theory* [25], or the well-known *HoTT Book* [26]. Though insufficient for understanding this dissertation, Escardó has a concise self-contained introduction to the univalence axiom itself [7]. Knowledge of basic dependent type theory is assumed from here on out.

*Convention* 2.1. As in the UniMath [10] formalization, we work in Martin-Löf type theory [19] with intensional identity types and no general W-types. Unlike the UniMath formalization, we do not assume in this text the existence of universes (Definition 2.10) unless stated. We freely assume function extensionality.

*Notation* 2.2. Write $\prod_{x:B} E[x]$ for dependent products, and $\sum_{x:B} E[x]$ and for dependent sums; $A \times B$ for binary products, and $A \amalg B$ for binary sums; $A \to B$ for non-dependent function types.

### Identification Types

Perhaps the most important aspect of univalent foundations is its treatment of equality between two two terms $a, b : A$ or types $A, B$ $\mathsf{type}$. There are two distinct notions of equality in Martin-Löf type theory.

The first notion is *judgemental equality*, written $A \equiv B$ $\mathsf{type}$ for types, $a \equiv b : A$ for terms, or simply $a \equiv b$ for either. Judgemental equality is symmetric, reflexive, transitive. It is also respected by all judgements: assuming judgemental equality $a \equiv b$, the type or term $a$ can be substituted throughout for $b$ during type checking. Judgemental equality includes computation ("beta") rules: for example $(\lambda x : B.\, a)\ b \equiv$

$a[x := b] : E[b]$ for $b : B$ and $x : B \vdash a : E[x]$; and uniqueness ("eta") rules: for example $f \equiv (\lambda x : B.\, f\ x) : \prod_{x:B} E[x]$. Judgemental equality also includes definitions: $\mathsf{xyz} \overset{\text{def}}{\equiv} a : A$ subsequently implies $\mathsf{xyz} \equiv a : A$.

*Convention* 2.3. Using the computation and uniqueness rules, we may write definitions in an implicit form, so that $a$ need not be given explicitly. For example, the function

$$\mathsf{curry} \colon (\mathbb{N} \times \mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N} \to \mathbb{N}$$

may be defined equivalently either in "closed form" as

$$\mathsf{curry} \overset{\text{def}}{\equiv} \lambda f\ x\ y.\, f\ \langle x, y \rangle\,, \tag{2.1}$$

or "implicitly" as

$$\mathsf{curry}\ f\ x\ y \overset{\text{def}}{\equiv} f\ \langle x, y \rangle\,. \tag{2.2}$$

The judgemental equation 2.2 follows from Equation 2.1 by computation rules, and the uniqueness of Equation 2.1 as defining Equation 2.2 follows by the uniqueness rules. It is left for the reader to infer the closed form of any implicit definition.

Judgemental equality is a **judgement**, not a type. Due to this, theorems in expressed in univalent foundations may not talk about judgemental equality. Instead, Martin-Löf type theory includes a second notion of equality: the *identification type*. For any type $A$ and terms $a, b : A$, the identification type is written $a =_A b$ or simply $a = b$. A term $p : a =_A b$ of identification type is sufficient to render $a$ and $b$ indistinguishable within the type theory using *identification induction* as discussed below.

The identification type comes equipped with a constructor $\mathsf{refl}_a : a =_A a$. Judgemental equality implies identification, in that $a \equiv b : A$ gives $\mathsf{refl}_a : a =_A b$. The converse, $p : a =_A b$ implying $a \equiv b : A$, is called *equality reflection*, and we do not allow it. Its absence is the marking characteristic of "intensional" dependent type theory.

Instead of equality reflection, the identification type permits *identification induction*. Following Convention 2.3, to define a function $f : \prod_{x,y:A} \prod_{z:x=_A y} E[x, y, z]$, we need only define the case $f\ x\ x\ \mathsf{refl}_x : E[x, x, \mathsf{refl}_x]$. This is justified by the eliminator $\mathsf{J}_{x,y,z.E[x,y,z]}$ with term formation rule

$$\frac{p : a =_A b \qquad r : \prod_{x:A} E[x, x, \mathsf{refl}_x]}{\mathsf{J}_{x,y,z.E[x,y,z]}(p, r) : E[a, b, p]} \tag{2.3}$$

and computation rule

$$\frac{r : \prod_{x:A} E[x, x, \mathsf{refl}_x]}{\mathsf{J}_{x,y,z.E[x,y,z]}(\mathsf{refl}_a, r) \equiv r\ a : E[a, a, \mathsf{refl}_a]}. \tag{2.4}$$

When identification induction is used, it is left for the reader to reconstruct the closed-form term using $\mathsf{J}$. More typically, given $p : a =_A b$ we will informally use substitution to replace occurrences of $a$ with $b$ or vice-versa, particularly when the specific $p$ is not relevant.

## Truncation Levels

In the logic of some foundations such as ZFC, the concepts of "propositions" and "sets" are distinct. The latter constitutes "substance" whereas the former describes it. By contrast, in univalent foundations it is the type that takes the role of both substance and its description.

The fundamental behaviour of a proposition over "substance" is that it is logically irrelevant. This is the concept known as *proof irrelevance*: though two proofs of a proposition may be syntactically different, the logic itself is unable to distinguish one from the other. In univalent foundations, we **define** propositions to be those types which satisfy proof irrelevance.

**Definition 2.4.** A type $A$ is a *proposition* (or *mere proposition* for emphasis) when for all elements $a, b : A$ there is a chosen identification $a =_A b$.

Being a proposition is the second-lowest class in a hierarchical classification of types called *truncation levels*. Truncation levels describe how trivial the type of identifications for a given type is. The lowest truncation level is the *contractible types*, the second truncation level is propositions, the third-lowest is *sets* or *0-type*s, and above that we have *n-type* for $n \geq 1$. Assuming function extensionality, it can be proven that "$A$ is of [a given truncation level]" is a proposition.

**Definition 2.5.** A type $A$ is *contractible* when there is a chosen element $\mathsf{pt} : A$ and for all elements $a : A$ there is a chosen identification $\mathsf{pt} =_A a$.

**Definition 2.6.** A type $A$ is a *set*[1] or *0-type* when $a =_A b$ is a proposition for all $a, b : A$; a *1-type* when $a =_A b$ is a set, and generally a $(k+1)$*-type* when $a =_A b$ is a $k$-type.

[1]This is orthogonal to matters of size. See Definition 2.11 for that.

**Examples 2.7.** The empty type `empty` is a proposition, and the unit type `unit` is contractible. The booleans `bool` and naturals $\mathbb{N}$ are both sets but not propositions. All contractible types are propositions, all propositions are sets, and all $k$-types are also $(k+1)$-types.

*Notation* 2.8. Write $\mathtt{iscontr}(A)$, $\mathtt{isaprop}(A)$ and $\mathtt{isaset}(A)$ for the types of $A$ respectively being contractible, a proposition, and a set. As a special case, when $A$ is a dependent sum $A \equiv \sum_{x:B} E[x]$ we shall write $\exists! x : B.\, E[x]$ for $\mathtt{iscontr}(\sum_{x:B} E[x])$.

**Definition 2.9.** A function $f\colon A \to B$ is an *equivalence*, written $\mathtt{isweq}(f)$,[2] when for all elements $y : B$, there exists a unique $x : A$ with an identification $f(x) =_B y$. That is, we have

$$\mathtt{isweq}(f) \stackrel{\text{def}}{\equiv} \prod_{y:B} \exists! x : A.\, f\ x =_B y$$

for the proposition of $f$ being an equivalence. Write $A \simeq B$ for the type of equivalences, that is

$$A \simeq B \stackrel{\text{def}}{\equiv} \sum_{f\colon A \to B} \mathtt{isweq}(f)\,.$$

Every equivalence $f : A \simeq B$ has a two-sided inverse $f^{-1} : B \simeq A$ satisfying

$$\prod_{x:A} f^{-1}(f(x)) = x \quad \text{and} \quad \prod_{y:B} f(f^{-1}(y)) = y\,.$$

Moreover, every function with a two-sided inverse may be upgraded to an equivalence.[3]

Sometimes the data of a type (which need not be a proposition) $A$ is not so important, only whether it is inhabited. Or to use set-theoretic terminology, whether such an element *exists*. The *propositional truncation* of $A$, written $\|A\|$, is the proposition which captures this idea. It comes with constructors

$$\frac{e : A}{|e| : \|A\|} \qquad \frac{e_1 : A \qquad e_2 : A}{|e_1| =_{\|A\|} |e_2|}\,, \tag{2.5}$$

[2]The name `isweq`, like most others in this section, comes from Voevodsky's original Foundations library. It may be considered short for "is a weak equivalence," although there is nothing weak about it.

[3]The type of "$f$ has a two-sided inverse" is not as well-behaved as the type "$f$ is an equivalence." Notably, the former fails to be a proposition for types that are not sets. This situation is similar to adjoint equivalences in category theory.

and for any proposition $P$ an eliminator

$$\frac{h : \|A\| \qquad f : A \to P}{\mathsf{u}_P(h, f) : P} \quad \text{satisfying} \quad \frac{t : A \qquad f : A \to P}{\mathsf{u}_P(|t|, f) \equiv f\ t : P}. \tag{2.6}$$

When $A$ is a dependent sum $A \equiv \sum_{x:B} E[x]$, we shall write $\exists x : B.\, E[x]$ for $\left\|\sum_{x:B} E[x]\right\|$.

## Universes

[TODO: say what universes are for. This section needs revision once I see the full scope of how many times I actually use universes. This turns out to be like once, maybe twice. I don't even end up using univalence for the main result, only function extensionality.]

**Definition 2.10.** A *universe* $(\mathcal{U}, \mathsf{El})$ is a type $\mathcal{U}$ of *codes* along with, for each element $A : \mathcal{U}$, a decoding $\mathsf{El}(A)$ $\mathsf{type}$. We shall often abuse notation and use simply $\mathcal{U}$ to refer to the universe.

**Definition 2.11.** For a universe $(\mathcal{U}, \mathsf{El})$, we say that a type $A$ is *$\mathcal{U}$-small* if there is an element $\mathsf{code}(A) : \mathcal{U}$ such that $\mathsf{El}(\mathsf{code}(A)) \equiv A$ $\mathsf{type}$. By contrast, a type is *large* with respect to some universes $\mathcal{U}, \mathcal{U}', \ldots$ when it is not small in any of them. We shall typically omit $\mathsf{El}$ and $\mathsf{code}$, so that $A : \mathcal{U}$ implies $A$ $\mathsf{type}$, and vice-versa when $A$ is $\mathcal{U}$-small.

*Convention* 2.12. We shall often simply say that a type is "small" or "large" when the universes it is small or large with respect to are obvious or have not given a name. This rule will also apply to other things that may be "small" or "large."

We will typically assume, unless otherwise mentioned, that a universe $\mathcal{U}$ is closed under all of the type constructors mentioned so far (products, sums, $\mathtt{unit}$, etc.). This holds of most of the universes we shall consider.

**Example 2.13.** Given a universe $(\mathcal{U}, \mathsf{El})$, there is a universe $(\mathtt{hSet}_{\mathcal{U}}, \mathsf{El}')$ of $\mathcal{U}$-small sets. The codes of $\mathtt{hSet}_{\mathcal{U}}$ are the pairs $\sum_{A:\mathcal{U}} \mathtt{isaset}(A)$, and the decoding is given by $\mathsf{El}'\langle A, h_A\rangle \stackrel{\text{def}}{\equiv} \mathsf{El}(A)$. Likewise, $\mathtt{hProp}_{\mathcal{U}}$ is the universe consisting of $\mathcal{U}$-small propositions $\sum_{A:\mathcal{U}} \mathtt{isaprop}(A)$, defined in much the same way. The universe $\mathtt{hProp}_{\mathcal{U}}$ is closed under dependent products, binary products, dependent sums, and identifications, but **not** binary sums. It also does not include $\mathtt{bool}$ or $\mathbb{N}$.

A *univalent universe* is a universe $\mathcal{U}$ where the canonical map

$$\begin{aligned} \texttt{id\_to\_weq} : \prod_{A,B:\mathcal{U}} (A =_{\mathcal{U}} B) \to (A \simeq B) \\ \texttt{id\_to\_weq}\ A\ A\ \mathsf{refl}_A \overset{\text{def}}{\equiv} \mathsf{id}_A \end{aligned} \tag{2.7}$$

given by identification induction is an equivalence for all types $A, B : \mathcal{U}$. For a given universe $\mathcal{U}$, the *univalence axiom* states that $\mathcal{U}$ is univalent. In short, the univalence axiom states that the identifications $A =_{\mathcal{U}} B$ between two types are characterized by the equivalences $A \simeq B$ between those types. These, in turn, depend on the structure of the identifications within $A$ and $B$ themselves.

**Example 2.14.** If a universe $\mathcal{U}$ is univalent, then so too is $\texttt{hSet}_{\mathcal{U}}$ and $\texttt{hProp}_{\mathcal{U}}$. The converse need not be true in general.

The type of equivalences between types, say $\texttt{bool} \simeq \texttt{bool}$, is not in general a proposition. Thus, the univalence axiom notably provides a way of constructing identifications which are provably **not** identifiable with $\mathsf{refl}$. This property is perfectly consistent with $\mathsf{J}$ [11]; indeed there are models including the univalence axiom that are as consistent as ZFC with some strongly inaccessible cardinals [12, 13].

## Classical Mathematics and Logic

By contrast to "univalent foundations," I use the term *set-based foundations*[4] to include both ZF(C) and type theories where all types are sets,[5] in the sense of Definition 2.6. Many theorems and definitions of set-based foundations may be recovered in univalent foundations by assuming certain types to be sets, such as by reasoning about $\texttt{hSet}_{\mathcal{U}}$ instead of $\mathcal{U}$ for a fixed universe $\mathcal{U}$.

It is an occasional misconception is that univalent foundations is incompatible with non-constructive reasoning such as the *law of excluded middle* (`LEM`) or the *axiom of choice* (`AxiomOfChoice`). These are perfectly admissible so long as they are stated correctly in terms of sets and propositions. However, they are often rendered unnecessary by using univalence: many choices become **unique** choices instead, and unique choices can be made constructively.

**Table 2.1.:** Example interpretations of mathematical English. $E[x]$ denotes a type indexed by $x$, and $P$ denotes a proposition.

| Mathematical English | Homotopy Type Theory | Homotopy Theory |
|---|---|---|
| Utterance | Judgement | Interpretation |
| *description of a **type** as below* | $A$ type | $A$ is a homotopy type |
| *desc. of a **type** mentioning* $x : B$ | $x : B \vdash E[x]$ type | $E$ is a fibre bundle over $B$ |
| *description of a **term*** | $t$ or $t : A$ | $t \in A$ |
| $t_1 : A$ is $t_2 : A$ | $t_1 \equiv t_2 : A$ | $t_1 = t_2$ |
| let $x$ be an $A$ | $\dfrac{x : A}{\dots}$ | let $x \in A$ |
| assume $P$ | $\dfrac{h : P}{\dots}$ | assume $P$ |
| **Definition** *xyz* is $t : A$. | $\mathsf{xyz} \overset{\text{def}}{\equiv} t : A$ | $\mathsf{xyz} \overset{\text{textupdef}}{=} t$ |
| **Theorem** $P$. *Proof.* $t$. $\square$ | $t : P$ | $t \in P$ |
| Description | Type | Space |
| [pair/triple/... of] $x : B$, $E[x]$ | $\sum_{x:B} E[x]$ | the total space $E$ |
| $x : B$ such that $P[x]$ | $\sum_{x:B} P[x]$ | subspace $\{\, x \in B \mid P[x] \text{ inhabited} \,\}$ |
| function $A$ to $B$ | $A \to B$ or $\prod_{x:A} B$ | function space $A \to B$ |
| for all $x : B$, $E[x]$ | $\prod_{x:B} E[x]$ or $\forall x : B.\, E[x]$ | sections of $E \to B$ |
| $A$ is a proposition | $\mathtt{isaprop}(A)$ | $A$ is contractible-if-inhabited |
| there exists a unique $A$ | $\mathtt{iscontr}(A)$ | $A$ is contractible |
| there exists a unique $x : B$ such that $P[x]$ | $\mathtt{iscontr}(\sum_{x:B} P[x])$ or $\exists! x : B.\, P[x]$ | |
| there exists $A$ | $\lVert A \rVert$ (propositional truncation) | $A$ is inhabited |
| there exists $x : B$ such that $P[x]$ | $\left\lVert \sum_{x:B} P[x] \right\rVert$ or $\exists x : B.\, P[x]$ | |
| $A$ is a set | $\mathtt{isaset}(A)$ | $A$ is a homotopy 0-type |
| $t_1 : A$ and $t_2 : A$ are identified | $t_1 =_A t_2$ | path space $t_1$ to $t_2$ |
| (if $A$ is a set) $t_1 : A$ and $t_2 : A$ are equal | $t_1 =_A t_2$ | path space $t_1$ to $t_2$ |
| either $A$ or $B$ | $A \amalg B$ | disjoint union of $A$ and $B$ |
| $A$ or $B$ | $A \vee B$ or $\lVert A \amalg B \rVert$ | |
| $f : A \to B$ is an equivalence | $\mathtt{isweq}(f)$ or $\prod_{y:B} \exists! x : A.\, f(x) =_B y$ | $f$ is a homotopy equivalence |
| $A$ and $B$ are equivalent | $A \simeq B$ or $\sum_{f:\, A \to B} \mathtt{isweq}(f)$ | $A$ and $B$ are homotopy equivalent |

## Further Conventions

*Convention* 2.15. Table 2.1 describes English language examples and their type-theoretic interpretation. In particular, we use a propositionally truncated interpretation of "there exists" and "or"; I use "there is" and "either … or" for the constructive variants. The terms "merely" and "chosen" may also be used respectively for emphasis. In addition to the informal English language interpretations, the final column suggests homotopical interpretations. These may be realized more formally by models of univalent foundations such as Voevodsky's in simplicial sets [12, 13].

*Convention* 2.16. I use upper case for types ($A, B, C, E, P$), calligraphic font for categories and the like ($\mathcal{C}, \mathcal{D}, \mathcal{N}, \mathcal{P}$), lower case for objects and morphisms within categories ($a, b : \mathcal{C},\ f, g \colon a \to b$), upper case for functors ($F, G, H \colon \mathcal{C} \to \mathcal{D}$), and Greek letters for natural transformations ($\alpha, \beta \colon F \Rightarrow G$).

*Notation* 2.17. Where a type contains a binary operator such as $\to$, $=$, or $\simeq$, we may use the infix notation $a \xrightarrow{f} b$ to mean $f \colon a \to b$. Sharing expressions in such infix notation, as in $t_1 \overset{p}{=} t_2 \overset{q}{=} t_3$, denotes the independent judgements $p : t_1 = t_2$ and $q : t_2 = t_3$. It may additionally denote the appropriate composition of $p$ and $q$ of the type $t_1 = t_3$ too, depending on the context.

*Convention* 2.18. A word or phrase written in *italics* is a technical term introduced at that point, and may be found in the index. When a theorem or definition has been formalized in UniMath, we shall write the name of its definition in `monospace` font, hyperlinked to the file and line number in the UniMath repository. An example is "a *category* (`category`) is … ." In anonymized versions, these links will necessarily be broken.

*Convention* 2.19. Definitions and theorems may be annotated in the margin with an icon to indicate their novelty.

- means a definition or theorem already present in UniMath.
- means a definition or theorem that is obvious or due to another author, but that has been newly formalized in UniMath as part of this dissertation.
- ★ means novel[6] definitions, statements, or results.

---

[4]Or "set-based mathematics?" I would like a term to capture basically "Lean and ZFC."
[5]This includes that of the *Lean 4 Theorem Prover* [22].
[6]As far as the I am aware, of course.

## 2.3. ROUGH Univalent Categories

Beyond the foundation of mathematics in which we work, the content of this dissertation is about non-associative generalizations of categories. Namely, Munch-Maccagnoni's *duploids* [23]. Before we can talk about univalent duploids, we ought understand categories in the context of univalent foundations.[7]

The authoritative paper on categories in univalent foundations is by Ahrens et al. [1], whose formalisation in ROCQ is today part of UNIMATH. An expanded version of that paper is included as Chapter 9 of the *HoTT Book* [26]. The development has since diverged from the paper: we use the names "`category`" and "`univalent_category`" for what used to be "precategory" and "category" respectively.

It is well-known that in a category $\mathcal{C}$, the correct notion of "sameness" between two objects $a, b : \mathcal{C}$ is *isomorphism* $a \cong b$, and not (necessarily) the foundations' notion of "equality" $a = b$. In set-based foundations, equality is overly restrictive: there may be objects $a \cong b$ that are not equal,[8] and there may be isomorphisms $a \cong a$ that are not the identity.[9] In univalent foundations, the type of identifications $a =_{\mathtt{ob}\,\mathcal{C}} b$ may very well correspond to isomorphisms $a \cong b$. A *univalent category* is one where they do.

An informative use-case of univalence is in relation to weak equivalences between categories. It is a theorem that for a univalent category $\mathcal{C}$ and a category $\mathcal{D}$, a fully-faithful and essentially surjective functor $F : \mathcal{C} \to \mathcal{D}$ is an adjoint equivalence. The analogous statement in set-based foundations amounts to the axiom of choice. More generally, objects determined by a universal property in a univalent category $\mathcal{C}$ become literally unique, as do left- and right- adjoints of a fixed functor $F : \mathcal{C} \to \mathcal{D}$ to another category $\mathcal{D}$. In other words, universal properties become mere properties and, when they hold, contractible. Thus, univalence has the practical benefit of being able to talk about *the* object (functor, category, etc.) satisfying a universal property, with the representative being canonical and choice-free.[10]

When the univalence axiom holds, many naturally-arising categories are univalent. For example, for a univalent universe $\mathcal{U}$, the category $\mathbf{Set}_{\mathcal{U}}$ (or simply $\mathbf{Set}$) of $\mathcal{U}$-small sets $A, B : \mathtt{hSet}_{\mathcal{U}}$ and functions $A \to B$ is univalent (`HSET_univalent_category`). As an extension, it was shown by Coquand et al. [6] that the univalence axiom extends to a "large class of algebraic structures." This implies that categories of set-based algebraic

[7] TODO: This paragraph is slightly out of place now that I have moved things around.
[8] Say, the sets $\{0, 1\}$ and $\{1, 2\}$.
[9] Say, the set $\mathbb{B}$ under boolean negation.
[10] TODO: How much does this even matter lol

structures such as monoids and rings are univalent too.

[TODO: This could possibly go, since I don't use the Rezk completion.] Every category $\mathcal{C}$ is weakly equivalent to a univalent category $\mathfrak{R}\mathcal{C}$. This is in the sense that there is a fully-faithful and essentially surjective functor $\mathfrak{r}_{\mathcal{C}} \colon \mathcal{C} \to \mathfrak{R}\mathcal{C}$. We call the category $\mathfrak{R}\mathcal{C}$ the *Rezk completion* of $\mathcal{C}$. One way to construct the Rezk completion is as the image of the Yoneda embedding $\mathsf{y} \colon \mathcal{C} \to [\mathcal{C}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}]$ [1].

The univalence of a category is sensitive to how the category is constructed. A relevant example is the *Kleisli category* of a monad

$$(\mathcal{C} \xrightarrow{T} \mathcal{C}, \quad \mathsf{Id}_{\mathcal{C}} \overset{\eta}{\Rightarrow} T, \quad T^2 \overset{\mu}{\Rightarrow} T)$$

on a category $\mathcal{C}$. One way to construct the Kleisli category of $T$ is as follows. The objects are those of $\mathcal{C}$, and, for objects $a, b : \mathcal{C}$, the morphisms are the Kleisli arrows $a \to Tb$ in $\mathcal{C}$. Identities and composition are through $\eta$ and $\mu$ respectively. Even when $\mathcal{C}$ is univalent, this construction of the Kleisli category is **not** generally univalent (see Example 2.20). Let us therefore call this non-univalent Kleisli category $\mathbf{Kleisli}_{\cancel{\mathsf{univ}}}(\mathcal{C}, T)$. There is a univalent construction of the Kleisli category, however, as the subcategory of free $T$-algebras in the Eilenberg-Moore category of $T$ (`univalent_kleisli_cat`). We shall call this category $\mathbf{Kleisli}_{\mathsf{univ}}(\mathcal{C}, T)$, or simply $\mathbf{Kleisli}(\mathcal{C}, T)$. Explicitly, this amounts to considering a suitable subcategory of $\mathcal{C}$ whose objects are of the form $Ta$ for some $a : \mathcal{C}$.

**Example 2.20** (Levy [17])**.** Consider the state monad $Ta \overset{\text{def}}{\equiv} (\mathbb{N} \Rightarrow a \times \mathbb{N})$ on $\mathbf{Set}$. Then the Kleisli arrows $a \to Tb$ of $T$ correspond to set-functions $a \times \mathbb{N} \to b \times \mathbb{N}$. From this it is easy to see that there is an isomorphism `unit` $\cong$ `bool` in $\mathbf{Kleisli}_{\cancel{\mathsf{univ}}}(\mathbf{Set}, T)$: this is simply an isomorphism `unit` $\times\, \mathbb{N} \cong$ `bool` $\times\, \mathbb{N}$ in $\mathbf{Set}$. However, the type `unit` is certainly not identified with `bool`!

# 3. ROUGH Unital Magmoids

## 3.1. Definitions

**Definition 3.1** (`category` [1])**.** A *category* $\mathcal{C}$ consists of the following:

1. A type of *objects*, written $\mathsf{ob}\,\mathcal{C}$ or just $\mathcal{C}$.
2. For each pair of objects $a, b : \mathcal{C}$, a type of *morphisms* from $a$ to $b$, also known as a *hom-type*, written $\mathcal{C}[\![a, b]\!]$ or $a \to b$.
3. For each object $a : \mathcal{C}$, a distinguished *identity* morphism $\mathsf{id}_a : \mathcal{C}[\![a, a]\!]$.
4. A *composition* operator $\fatsemi_{\mathcal{C}}$ or simply $\fatsemi$ assigning to each pair of morphisms $a \xrightarrow{f} b \xrightarrow{g} c$ a composite $a \xrightarrow{f \fatsemi g} c$ such that:
   a) Composition is *unital*: for $a \xrightarrow{f} b$ we have $\mathsf{id}_a \fatsemi f = f$ and $f \fatsemi \mathsf{id}_b = f$.
   b) Composition is *associative*: for $a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$ we have $f \fatsemi (g \fatsemi h) = (f \fatsemi g) \fatsemi h$.
5. $\mathcal{C}$ *has hom-sets*: each hom-type $\mathcal{C}[\![a, b]\!]$ is a set.

Most of Definition 3.1 will be familiar to a mathematician who has come across categories. Indeed, putting aside the hom-set law for now (Definition 3.1.5), one obtains a typical set-theoretic definition by substituting "class" for "type" in the definition of objects and morphisms. The hom-set law simply makes identitification $f =_{a \to b} g$ of morphisms $f, g \colon a \to b$ be a proposition. We shall therefore find it convenient to use the language of "equality" for identifications between morphisms.

**Definition 3.2.** A *unital magmoid* or *non-associative category* [18] (`unital_magmoid`) is a generalization of a category that need not be associative (Definition 3.1.4.b). Explicitly, a unital magmoid $\mathcal{M}$ consists of the same data as a category—objects $\mathsf{ob}\,\mathcal{M}$, hom-types $\mathcal{M}[\![a, b]\!]$, identities $\mathsf{id}_a$, and composition $f \fatsemi g$—such that composition is unital and all hom-types are sets.

We import most definitions of structures in and on categories for unital magmoids directly, as much as they are well-defined. I spell out these definitions here in their entirety. The failure of associativity causes the imported notions to become weaker, however. We shall see this for isomorphisms (Definition 3.9) and natural transformations (Examples 3.16).

**Definition 3.3.** Let $\mathcal{U}$ be a universe. A unital magmoid or category $\mathcal{M}$ is *locally $\mathcal{U}$-small* if for all $a, b : \mathcal{M}$ the hom-type $\mathcal{M}[\![a, b]\!]$ is $\mathcal{U}$-small. A unital magmoid or category $\mathcal{M}$ is *$\mathcal{U}$-small* when it is locally small and its type of objects $\mathsf{ob}\,\mathcal{M}$ is $\mathcal{U}$-small. A unital magmoid or category is *large* with respect to some universes $\mathcal{U}, \mathcal{U}', \ldots$ when it is not small in any of them. Following Convention 2.12, we shall often omit the universe when describing unital magmoids and categories as "small" or "large."

**Example 3.4.** For a unital magmoid or category $\mathcal{M}$, its *opposite* $\mathcal{M}^{\mathsf{op}}$ is the unital magmoid or category with the same objects, and with the directions of all morphisms inverted:

$$\mathsf{ob}\,\mathcal{M}^{\mathsf{op}} \overset{\text{def}}{\equiv} \mathsf{ob}\,\mathcal{M}$$
$$\mathcal{M}^{\mathsf{op}}[\![a, b]\!] \overset{\text{def}}{\equiv} \mathcal{M}[\![b, a]\!]\,.$$

The identity $\mathsf{id}_a$ for $a : \mathcal{M}^{\mathsf{op}}$ is that of $\mathcal{M}$, and composition of two morphisms $f : \mathcal{M}^{\mathsf{op}}[\![a, b]\!]$ and $g : \mathcal{M}^{\mathsf{op}}[\![b, c]\!]$ is given by composition in $\mathcal{M}$:

$$f \mathbin{⨾}_{\mathcal{M}^{\mathsf{op}}} g \overset{\text{def}}{\equiv} g \mathbin{⨾}_{\mathcal{M}} f\,.$$

The unital magmoid and category laws of $\mathcal{M}^{\mathsf{op}}$ all follow from those of $\mathcal{M}$.

**Definition 3.5** (`is_z_isomorphism`, `z_iso`)**.** Let $\mathcal{M}$ be a unital magmoid or category. A morphism $f\colon a \to b$ between objects $a, b : \mathcal{M}$ *has an inverse* (or *two-sided-inverse* for emphasis) if there is a chosen morphism $f^{-1} : b \to a$ such that

$$f \mathbin{⨾} f^{-1} = \mathsf{id}_a \quad \text{and} \quad f^{-1} \mathbin{⨾} f = \mathsf{id}_b\,. \tag{3.1}$$

In this case, we say that $f$ is an *isomorphism*, written $f : a \cong b$.

*Remark* 3.6 (`isaprop_is_z_isomorphism`). It is well known that inverses of morphisms are unique in a category. That is, if we have a morphism $f\colon a \to b$ in a category $\mathcal{C}$ and any two morphisms $p, q\colon b \to a$ satisfying Equation 3.1, then we have $p = q$. I spell out the proof, because it depends on associativity:

$$p = p \fatsemi (f \fatsemi q) = (p \fatsemi f) \fatsemi q = q\,.$$

In combination with the hom-set law, this makes the type "$f$ has an inverse" be a proposition in a category (`isaprop_is_z_isomorphism`).

**Definition 3.7** (`is_univalent` [1])**.** Let $\mathcal{C}$ be a category. Then there is a canonical map from identifications to isomorphisms

$$\begin{aligned} \texttt{id\_to\_iso} &: \prod_{a,b:\mathsf{ob}\,\mathcal{C}} (a =_{\mathsf{ob}\,\mathcal{C}} b) \to (a \cong b) \\ \texttt{id\_to\_iso}\ a\ a\ \mathsf{refl}_a &\stackrel{\text{def}}{\equiv} \mathsf{id}_a \end{aligned} \tag{3.2}$$

defined by identification induction (reminiscent of Equation 2.7). The category $\mathcal{C}$ is *univalent* if `id_to_iso` $a\ b$ is an equivalence for all $a, b : \mathcal{C}$.

**Lemma 3.8** (`is_rxgraph_univalent_from_isaprop_edges_from` [25, Chapter 11])**.** *In a category $\mathcal{C}$, for an object $a : \mathcal{C}$, the type $\sum_{b:\mathcal{C}} a \cong b$ is called the* fan *of $a$. The following are equivalent.*

1. *The category $\mathcal{C}$ is univalent.*
2. *For every object $a : \mathcal{C}$, the fan of $a$ is a proposition.*
3. *For every object $a : \mathcal{C}$, the fan of $a$ is contractible with centre $\langle a, \mathsf{id}_a\rangle : \sum_{b:\mathcal{C}} a \cong b$.*

*Proof.* This is a variant of Rijke's *fundamental theorem of identity types* [25, Chapter 11]. The formalization and Rijke both make a more general statement about "reflexive graphs" and "identity systems" respectively, but we do not develop the language for those statements in this dissertation. □

**Definition 3.9.** In a unital magmoid, we still use the term isomorphism and its notation $\cong$ to mean a morphism that has a specified inverse, even though such isomorphisms are inferior to those in categories. For example, inverses are no longer necessarily unique (see Example A.2). Thus we do not generalize the definition of "univalent category" to unital magmoids directly. Instead, we shall define a stronger notion of isomorphism and use that for the univalence condition of unital magmoids (see Definition 3.29), which simplifies to Definition 3.7 when the unital magmoid is a category.

**Definition 3.10** (`functor`)**.** A *functor* (or *categorical functor* for emphasis) $F$ between categories or unital magmoids $\mathcal{N}$ and $\mathcal{M}$, written $F : \mathcal{N} \to \mathcal{M}$, consists of:

1. A mapping on objects $F_0 \colon \mathsf{ob}\,\mathcal{N} \to \mathsf{ob}\,\mathcal{M}$.

2. For all objects $a, b : \mathsf{ob}\,\mathcal{N}$, a mapping on morphisms $F_1 : \mathcal{N}[\![a, b]\!] \to \mathcal{N}[\![F_0\,a, F_0\,b]\!]$.

3. $F$ preserves:

   a) Identities:
   $$F_1\left(a \xrightarrow{\mathsf{id}_a} a\right) = F_0\ a \xrightarrow{\mathsf{id}_{F_0 a}} F_0\,a$$
   for all $a : \mathcal{N}$.

   b) Composition:
   $$F_1\left(a \xrightarrow{f} b \xrightarrow{g} c\right) = F_0\,a \xrightarrow{F_1\,f} F_0\,b \xrightarrow{F_1\,g} F_0\,c$$
   for all $a \xrightarrow{f} b \xrightarrow{g} c$ in $\mathcal{N}$.

Both $F_0$ and $F_1$ may simply be written $F$.

**Definition 3.11** (rxfunctor)**.** A *reflexive graph functor* $F$ between unital magmoids or categories $\mathcal{N}$ and $\mathcal{M}$, written $F \colon \mathcal{N} \dashrightarrow \mathcal{M}$, consists of the same data as a categorical functor, but need only preserve identities.

**Example 3.12.** Let $\mathcal{M}$ be a unital magmoid. There is a categorical functor $\mathsf{Id}_{\mathcal{M}} \colon \mathcal{M} \to \mathcal{M}$ which is the identity on objects and on morphisms: $F_0\,a \overset{\text{def}}{\equiv} a$ and $F_1\,f \overset{\text{def}}{\equiv} f$. Let $\mathcal{M}_1$, $\mathcal{M}_2$ and $\mathcal{M}_3$ be unital magmoids, and $\mathcal{M}_1 \overset{F}{\dashrightarrow} \mathcal{M}_2 \overset{G}{\dashrightarrow} \mathcal{M}_3$ be categorical or reflexive graph functors between them. Then there is a composite reflexive graph or categorical functor $GF \colon \mathcal{M}_1 \dashrightarrow \mathcal{M}_3$ or $GF \colon \mathcal{M}_1 \to \mathcal{M}_3$. [TODO: finish this.]

**Definition 3.13** (fully_faithful)**.** A categorical or reflexive graph functor $F \colon \mathcal{N} \dashrightarrow \mathcal{M}$ is *fully-faithful* if its action on morphisms $F_1 : \mathcal{N}[\![a, b]\!] \to \mathcal{M}[\![Fa, Fb]\!]$ is an equivalence for all $a, b : \mathcal{N}$.

**Definition 3.14** (nat_trans)**.** A *natural transformation* $\alpha$ between categorical or reflexive graph functors $F, G \colon \mathcal{N} \to \mathcal{M}$, written $\alpha \colon F \Rightarrow G$, consists of:

1. For each object $a : \mathcal{N}$, a morphism $\alpha_a \colon Fa \to Ga$.

2. *Naturality*: for all objects $a, b : \mathcal{N}$ and morphisms $f \colon a \to b$, the equation

$\alpha_a \fatsemi Gf = Ff \fatsemi \alpha_b$ holds. That is, the *naturality square*

$$\begin{tikzcd} Fa \arrow[r, "Ff"] \arrow[d, "\alpha_a"'] & Fb \arrow[d, "\alpha_b"] \\ Ga \arrow[r, "Gf"] & Gb \end{tikzcd} \tag{3.3}$$

in $\mathcal{M}$ commutes.

**Definition 3.15.** Let $F, G : \mathcal{N} \dashrightarrow \mathcal{M}$ be categorical or reflexive graph functors. We call the family of morphisms $\prod_{a:\mathcal{N}} Fa \to Ga$ that need not satisfy the naturality condition an *unnatural transformation* and write $F \rightharpoonup G$ for its type.

**Examples 3.16.** Let $F : \mathcal{N} \dashrightarrow \mathcal{M}$ be a categorical or reflexive graph functor. There is a natural transformation $\mathsf{id}_F : F \Rightarrow F$ given componentwise by $(\mathsf{id}_F)_a \overset{\text{def}}{\equiv} \mathsf{id}_{Fa}$, where naturality follows by unitality (Definition 3.1.4.a). For graph functors $F, G, H : \mathcal{N} \rightharpoonup \mathcal{M}$ and natural transformations $\alpha : F \Rightarrow G$, and $\beta : G \Rightarrow H$, we may define their vertical composite $(\alpha \fatsemi \beta)_a \overset{\text{def}}{\equiv} \alpha_a \fatsemi \beta_a$. However, this is only an unnatural transformation $\alpha \fatsemi \beta : F \rightharpoonup H$ in general, due to the non-associativity of $\mathcal{M}$. When $\mathcal{M}$ is a category, the vertical composite $\alpha \fatsemi \beta$ is natural.

**Example 3.17.** Let $\mathcal{C}$ and $\mathcal{D}$ be categories with functors $F, G : \mathcal{C} \to \mathcal{D}$ between them. Then the type of natural transformations $F \Rightarrow G$ is a set because $\mathcal{D}$ has hom-sets. Thus, there is a category $[\mathcal{C}, \mathcal{D}]$ (`functor_category`) whose objects are functors $F, G : \mathcal{C} \to \mathcal{D}$ and whose morphisms are natural transformations $\alpha : F \Rightarrow G$. If $\mathcal{D}$ is univalent, then the functor category $[\mathcal{C}, \mathcal{D}]$ is univalent. That is, the identifications $F = G$ between two functors $F, G : \mathcal{C} \to \mathcal{D}$ correspond to the natural isomorphisms $F \cong G$ between them (`is_univalent_functor_category`).

**Non-example 3.18.** Let $\mathcal{U}$ be a universe. There is in general no category $\mathbf{Cat}_{\mathcal{U}}$ of $\mathcal{U}$-small categories and functors between them. This is because, in general, the functors $\mathcal{C} \to \mathcal{D}$ between categories $\mathcal{C}$ and $\mathcal{D}$ do not form a set.[1] One must instead consider $\mathbf{Cat}_{\mathcal{U}}$ as a bicategory. Another approach, more reminiscent of set-based foundations, is to restrict ourselves to *strict categories*, or $\mathtt{hSet}_{\mathcal{U}}$-small categories when $\mathcal{U}$-small. These are categories $\mathcal{C}$ whose objects $\mathsf{ob}\,\mathcal{C}$ form a set. Then we may perfectly well consider the category $\mathbf{Cat}_{\mathtt{hSet}_{\mathcal{U}}}$ of strict $\mathcal{U}$-small categories.

[1]Consider the two natural isomorphisms from $\mathbf{1} \xrightarrow{\mathtt{bool}} \mathbf{Set}$ to itself, when $\mathbf{Set}$ is univalent.

## 3.2. Polarities

Many theorems about categories which fail to generalize to unital magmoids can be salvaged by hypothesizing associativity of certain composites.

**Definition 3.19** (is_thunkable, is_negative, etc. [23])**.** We say that a path of morphisms

$$a_0 \xrightarrow{f_1} a_1 \xrightarrow{f_2} a_2 \to \cdots \to a_{n-1} \xrightarrow{f_n} a_n \tag{3.4}$$

in a unital magmoid *associates* when all ways of parenthesising their composite $f_1 \fatsemi f_2 \fatsemi \cdots \fatsemi f_n$ ($= \mathsf{id}_{a_0}$ if empty) are equal. In the simplest non-trivial case, a triple of morphisms $a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$ associates if and only if we have $f \fatsemi (g \fatsemi h) = (f \fatsemi g) \fatsemi h$.

We say that:

- $f$ is *thunkable* when all triples $(f, g, h)$ associate.
- $h$ is *linear* when all triples $(f, g, h)$ associate.
- $g$ is *intermediate* when all triples $(f, g, h)$ associate.
- $b$ is (*semantically*) *negative* when all morphisms into $b$ are thunkable.
- $c$ is (*semantically*) *positive* when all morphisms out of $c$ are linear.

This is depicted pictorially in Figure 3.1.

intermediate

thunkable linear

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

negative positive

**Figure 3.1.:** The triple $(f, g, h)$ associates whenever any of the labelled properties holds.

*Notation* 3.20. When a path of morphisms as in Diagram 3.4 is known to associate, we will occasionally write its composite without parentheses as $f_1 \fatsemi f_2 \fatsemi \cdots \fatsemi f_n$. Occasionally, following Notation 2.17, we will use the diagram $a_0 \xrightarrow{f_1} a_1 \xrightarrow{f_2} a_2 \to \cdots \to a_{n-1} \xrightarrow{f_n} a_n$ itself to denote the composite.

*Remark* 3.21. It is easy to see that thunkability/linearity and negativity/positivity are dual, while intermediateness is self-dual. This is in the usual sense that being thunkable in some unital magmoid $\mathcal{C}$ is the same as being thunkable in $\mathcal{C}^{\mathsf{op}}$, and being negative in $\mathcal{C}$ is the same as being positive in $\mathcal{C}^{\mathsf{op}}$, up to inversion of the identification $f \mathbin{;} (g \mathbin{;} h) = (f \mathbin{;} g) \mathbin{;} h$.

**Example 3.22.** In any unital magmoid, the identity morphisms are thunkable, linear, and intermediate by unitality (Definition 3.1.4.a). Composites of thunkable, linear, or intermediate morphisms are also thunkable, linear, or intermediate respectively.

**Definition 3.23** (`linear_category`, `negative_category`, etc. [23])**.** Let $\mathcal{M}$ be a unital magmoid. The following are subcategories of $\mathcal{M}$; that is, sub- unital magmoids of $\mathcal{M}$ which are associative.

- The *linear subcategory* $\mathcal{M}_l$, restricting to linear morphisms.
- The *thunkable subcategory* $\mathcal{M}_t$, restricting to thunkable morphisms.
- The *intermediate subcategory* $\mathcal{M}_i$, restricting to intermediate morphisms.
- The (*semantically*) *negative subcategory* $\mathcal{M}^{\ominus}$, restricting to semantically negative objects.
- The (*semantically*) *positive subcategory* $\mathcal{M}^{\oplus}$, restricting to semantically positive objects.
- Combinations of the above, notably:
  - The *linear-and-thunkable subcategory* $\mathcal{M}_{lt}$, restricting to linear-and-thunkable morphisms.
  - The *linear-and-thunkable-and-intermediate subcategory* $\mathcal{M}_{lti}$.
  - The *negative-linear subcategory* $\mathcal{M}_l^{\ominus}$, the full subcategory of $\mathcal{M}_l$ of semantically negative objects.
  - The *positive-thunkable subcategory* $\mathcal{M}_t^{\oplus}$, the full subcategory of $\mathcal{M}_t$ of semantically negative objects.

By the property of polarization, all morphisms of $\mathcal{M}^{\ominus}$ and $\mathcal{M}_l^{\ominus}$ are thunkable and intermediate, and all morphisms of $\mathcal{M}^{\oplus}$ and $\mathcal{M}_t^{\oplus}$ are linear and intermediate. Figure 3.2 depicts some inclusion functors between the subcategories.

$$\begin{tikzcd}
\mathcal{M}^\oplus \arrow[r, hook, two heads] & \mathcal{M}_l & \\
\mathcal{M}_t^\oplus \arrow[u, hook] \arrow[r, hook, two heads] \arrow[dr, hook, two heads, bend right] & \mathcal{M}_{lt} \arrow[u, hook] \arrow[d, hook] & \mathcal{M}_l^\ominus \arrow[l, hook', two heads] \arrow[ul, hook', two heads, bend right] \arrow[d, hook] \\
& \mathcal{M}_t & \mathcal{M}^\ominus \arrow[l, hook', two heads]
\end{tikzcd}$$

**Figure 3.2.:** Inclusion functors between the subcategories of a unital magmoid $\mathcal{M}$. The double arrow head ($\twoheadrightarrow$) means full, and the hook ($\hookrightarrow$) means faithful.

*Notation* 3.24. Let $a, b : \mathcal{M}$ be objects in a unital magmoid $\mathcal{M}$. We may write the subscripts from Definition 3.23 onto the binary operators of morphisms $a \to b$ or isomorphisms $a \cong b$ to denote that the morphism or isomorphism resides in the corresponding subcategory of the unital magmoid. For example, $a \to_l b$ denotes a linear morphism, and $a \cong_{lt} b$ denotes a *linear-and-thunkable isomorphism*: an isomorphism which is linear-and-thunkable, and has a linear-and-thunkable inverse.

## 3.3. Univalence

**Lemma 3.25** (`i_iso_interpose`)**.** *In a unital magmoid, for an intermediate isomorphism $p : b \cong_i b'$, the equation $(f \mathbin{⨾} p) \mathbin{⨾} (p^{-1} \mathbin{⨾} g) = f \mathbin{⨾} g$ holds for all $a \xrightarrow{f} b \xrightarrow{g} c$.*

*Proof.* The path $a \xrightarrow{f} b \xrightarrow{p}_i b' \xrightarrow{p^{-1}}_i b \xrightarrow{g} c$ associates by intermediateness of $p$ and $p^{-1}$. Thus we have

$$\begin{aligned}\left(a \xrightarrow{f} b \xrightarrow{p}_i b'\right) ⨾ \left(b' \xrightarrow{p^{-1}}_i b \xrightarrow{g} c\right) &= a \xrightarrow{f} \left(b \xrightarrow{p}_i b' \xrightarrow{p^{-1}}_i b\right) \xrightarrow{g} c \\ &= a \xrightarrow{f} b \xrightarrow{g} c\,.\end{aligned}$$

□

**Lemma 3.26** (`is_intermediate_from_interpose`)**.** *In a unital magmoid, if $p : b \cong b'$ is an isomorphism with thunkable inverse $p^{-1} \colon b' \to_t b$, and $p$ satisfies $(f ⨾ p) ⨾ (p^{-1} ⨾ g) = f ⨾ g$ for all $a \xrightarrow{f} b \xrightarrow{g} c$, then $p \colon b \to b'$ is intermediate.*

*Proof.* Fix $f \colon a \to b$ and $g \colon b' \to c$. We need to show $f ⨾ (p ⨾ g) = (f ⨾ p) ⨾ g$. Indeed, we have

$$a \xrightarrow{f} \left(b \xrightarrow{p} b' \xrightarrow{g} c\right) = \left(a \xrightarrow{f} b \xrightarrow{p} b'\right) \fatsemi \left(b' \xrightarrow{p^{-1}}_t b \xrightarrow{p} b' \xrightarrow{g} c\right)$$
$$= \left(a \xrightarrow{f} b \xrightarrow{p} b'\right) \xrightarrow{g} c\,.$$

□

★ *Remark* 3.27. Linear-and-thunkable isomorphisms are **not** always intermediate in a unital magmoid (see Example A.1 for a counterexample). However, they are in the presence of positive or negative objects (see Lemma 4.13).

★ **Lemma 3.28.** *In a unital magmoid, the linear-and-thunkable-and-intermediate isomorphisms $a \cong_{lti} b$ correspond to the indiscernibilities $a \asymp b$ as defined by Ahrens et al. [2].*

*Proof (sketch).* This follows from the previous lemmas (3.25, 3.26, TODO Yoneda). Note that the structure of unital magmoids does not differ from that of a category. □

★ **Definition 3.29.** A unital magmoid $\mathcal{M}$ is *univalent* if the linear-and-thunkable-and-intermediate subcategory $\mathcal{M}_{lti}$ is univalent (Definition 3.7).

**Definition 3.30.** A functor $F \colon \mathcal{M} \to \mathcal{M}'$ between unital magmoids $\mathcal{M}$ and $\mathcal{M}'$ that is:

- A *strong equivalence* if it is fully-faithful, and for every object $b : \mathcal{M}'$, there is a chosen object $a : \mathcal{M}$ and linear-and-thunkable-and-intermediate isomorphism $Fab_{lti}$ in $\mathcal{M}'$.
- A *weak equivalence* if it is likewise fully-faithful, but for all $b : \mathcal{M}'$ the object $a : \mathcal{M}$ and linear-and-thunkable-isomorphism $Fab_{lti}$ only merely exist.

*Remark* 3.31. It follows from Lemma 3.28 and [Theorem 6.4/17.11 2, 2025] [2] that strong equivalences of unital magmoids correspond to identifications between those unital magmoids.

# 4. ROUGH Duploids

## 4.1. Definitions and Properties

**Definition 4.1** (`preduploid` [23, 5, 18])**.** A *preduploid* (or *single-sorted preduploid* for emphasis) is a unital magmoid such that each object is semantically negative or semantically positive.

**Definition 4.2** (`duploid` [23, 5, 18])**.** A *duploid* (or *single-sorted duploid*) is a preduploid $\mathcal{D}$ equipped with *polarity shifts*:

1. (`has_negative_shifts`) For every object $a : \mathcal{D}$,
   a) its *upshift*, a semantically negative object written $\Uparrow a : \mathcal{D}$,
   b) a linear morphism $\texttt{force}_a : \Uparrow a \to_l a$,
   c) a linear-and-thunkable inverse for $\texttt{force}_a$ called $\texttt{delay}_a : a \to_{lt} \Uparrow a$.
2. (`has_positive_shifts`) Dually, for every object $a : \mathcal{D}$,
   a) its *downshift*, a semantically positive object written $\Downarrow a : \mathcal{D}$,
   b) a thunkable morphism $\texttt{wrap}_a : a \to \Downarrow a$,
   c) a linear-and-thunkable inverse for $\texttt{wrap}_a$ called $\texttt{unwrap}_a : \Downarrow a \to_{lt} a$.

*Remark* 4.3. Thunkability of each $\texttt{delay}_a$ and linearity of each $\texttt{unwrap}_a$ follow from the polarities of $\Uparrow a$ and $\Downarrow a$.

*Remark* 4.4. Definition 4.1 makes the polarities in a preduploid be a mere property on unital magmoids, rather than structure. This differs from the original construction by Munch-Maccagnoni [23] in which preduploids were defined in terms of separate negative and positive categories with additional mixed composition operators. I therefore call the duploids and preduploids from the original definition *two-sorted (pre)duploids* to distinguish them from the single-sorted variants defined above. We subsume two-sorted duploids and preduploids into the theory of their single-sorted counterparts by considering the *polarity mapping* as extra structure.

**Definition 4.5** (split_preduploid)**.** A *split preduploid* is a unital magmoid $\mathcal{D}$ equipped with a chosen *polarity mapping* $\omega\colon \mathsf{ob}\,\mathcal{D} \to \{\ominus, \oplus\}$. We say that an object $a$ is *syntactically negative* if $\omega(a) = \ominus$ and *syntactically positive* if $\omega(a) = \oplus$, and require that every **syntactically** negative/positive object is also **semantically** negative/positive respectively.

*Remark* 4.6 (polarity_mapping_to_has_polarities, polarity_mapping_from_LEM). Every split preduploid is evidently a single-sorted preduploid. Conversely, the law of the excluded middle allows defining a polarity mapping for any preduploid. For example, as

$$\omega(a) \overset{\text{def}}{\equiv} \begin{cases} \ominus & \text{if } a \text{ is semantically negative,} \\ \oplus & \text{otherwise.} \end{cases}$$

**Definition 4.7** (split_duploid)**.** Let $\mathcal{D}$ be a single-sorted duploid with a polarity mapping that makes it a split preduploid. Then $\mathcal{D}$ is a *split duploid* when $\Uparrow a$ and $\Downarrow a$ are also syntactically negative and positive respectively.

*Remark* 4.8. The original two-sorted duploids of Munch-Maccagnoni [23, 2014] are precisely the split duploids where $\Uparrow$/ `force` and $\Downarrow$/ `wrap` are trivial—that is, identities—for syntactically negative and syntactically positive objects respectively.

**Definition 4.9** (chosen_negative_category, …)**.** Let $\mathcal{D}$ be a split preduploid with polarity mapping $\omega\colon \mathsf{ob}\,\mathcal{D} \to \{\ominus, \oplus\}$. Extending Definition 3.23, there are the following subcategories of $\mathcal{D}$.

- The *syntactically negative subcategory* $\mathcal{D}^{\ominus\omega}$ is the full subcategory of $\mathcal{D}^{\ominus}$ containing only syntactically negative objects.
- The *syntactically positive subcategory* $\mathcal{D}^{\oplus\omega}$ is the full subcategory of $\mathcal{D}^{\oplus}$ containing only syntactically positive objects.
- Combinations with those of Definition 3.23, notably:
  - The *syntactically negative-linear subcategory* $\mathcal{D}_l^{\ominus\omega}$.
  - The *syntactically positive-thunkable subcategory* $\mathcal{D}_t^{\oplus\omega}$.

**Lemma 4.10** (is_thunkable_iff_delay_wrap, … [23])**.** *In a duploid, a morphism $f\colon a \to b$ is thunkable if and only if the triple*

$$a \xrightarrow{f} b \xrightarrow{\mathtt{delay}_b}_{lt} \Uparrow b \xrightarrow{\mathtt{wrap}_{\Uparrow b}}_{ti} \Uparrow\Downarrow b \quad \textit{associates.}$$

*Dually, $f\colon a \to b$ is linear if and only if the triple*

$$\Uparrow\Downarrow a \xrightarrow{\texttt{force}_{\Downarrow a}}_{li} \Downarrow a \xrightarrow{\texttt{unwrap}_a}_{lt} a \xrightarrow{f} b \quad \textit{associates.}$$

*Proof.* This is Proposition 13 of Munch-Maccagnoni, 2014 [23]. □

**Lemma 4.11** (`is_positive_of_lt_iso`)**.** *In a unital magmoid $\mathcal{M}$ with a linear-and-thunkable isomorphism $p : a \cong_{lt} b$, the object $a$ is positive if and only if $b$ is positive, and $a$ is negative if and only if $b$ is negative.*

*Proof.* Without loss of generality, we need only prove that $a$ being positive implies that $b$ is positive. By definition, this is that for any object $c$, any morphism $f\colon b \to c$ is linear. We have

$$\begin{aligned} b \xrightarrow{f} c &= \left(b \xrightarrow{p^{-1}}_{lt} a \xrightarrow{p}_{lt} b\right) \xrightarrow{f} c \\ &= b \xrightarrow{p^{-1}}_{lt} \left(a \xrightarrow{p}_{lt} b \xrightarrow{f} c\right) . \end{aligned}$$

This is linear because $p^{-1}$ is linear and $a$ is positive. □

**Lemma 4.12** (`lt_iso_upshift_of_negative`, …)**.** *For an object $a : \mathcal{D}$ in a duploid $\mathcal{D}$, the following are equivalent:*

1. *$a$ is negative.*
2. *$\texttt{force}_a\colon \Uparrow a \to_l a$ is thunkable.*
3. *$\texttt{force}_a$ is a linear-and-thunkable isomorphism $\Uparrow a \cong_{lt} a$, and $\texttt{delay}_a : a \to_{lt} \Uparrow a$ is its inverse.*

*Dually, for $a : \mathcal{D}$ again, the following are equivalent.*

1. *$a$ is positive.*
2. *$\texttt{wrap}_a : a \to_t \Downarrow a$ is linear.*
3. *$\texttt{wrap}_a$ is a thunkable-and-linear isomorphism $a \cong_{lt} \Downarrow a$, and $\texttt{unwrap}_a : \Downarrow a \to_{lt} a$ is its inverse.*

*Proof.* We have (3) ⇒ (1) by Lemma 4.11, and (1) ⇒ (2) by definition of the polarities. Finally, (2) is by definition the only thing missing to make (3) true, so we have (3) ⇒ (2). □

## 4.2. Univalence

**Lemma 4.13** (lt_iso_is_intermediate_from_polarized)**.** *In a unital magmoid $\mathcal{M}$, if an object $a : \mathcal{M}$ is positive or negative, then all linear-and-thunkable isomorphisms $p : a \cong_{lt} b$ are intermediate.*

*Proof.* If $a$ is negative, then $p$ is automatically intermediate. If $a$ is positive, then $b$ must be positive too by Lemma 4.11, and so $p$ is automatically intermediate. □

**Corollary 4.14** (preduploid_lt_iso_is_intermediate)**.** *Every linear-and-thunkable isomorphism $p : a \cong_{lt} b$ in a preduploid is intermediate.*

*Proof.* This follows from Lemma 4.13, since $a$ is positive or negative by the preduploid property (Definition 4.1). □

**Definition 4.15** (is_duploid_univalent)**.** A duploid or preduploid $\mathcal{D}$ is *univalent* (or *univalent as a single-sorted (pre)duploid* for emphasis) if its linear-and-thunkable subcategory $\mathcal{D}_{lt}$ is univalent.

*Remark* 4.16 (unital_magmoid_univalent_iff_duploid_univalent). By Corollary 4.14, this coincides with univalence of $\mathcal{D}$ as a unital magmoid.

**Theorem 4.17** (is_duploid_univalent_from_positive_thunkable_and_negative_-linear_categories)**.** *A preduploid $\mathcal{D}$ is univalent if and only if the subcategories $\mathcal{D}_t^{\oplus}$ and $\mathcal{D}_l^{\ominus}$ are univalent.*

*Proof.* The "only if" direction follows immediately from the fully-faithfulness of the inclusion functors $\mathcal{M}_t^{\oplus} \hookrightarrow\!\!\!\!\rightarrow \mathcal{M}_{lt} \twoheadleftarrow\!\!\!\hookleftarrow \mathcal{M}_l^{\ominus}$.

For the "if" direction, we apply Lemma 3.8 and show that for all $a : \mathcal{D}$ the type of fans $\sum_{b:\mathcal{D}} a \cong_{lt} b$ is a proposition. By the preduploid property, assume that $a$ is positive (dually, negative). Then fans $\sum_{b:\mathcal{D}} a \cong_{lt} b$ are equivalent to fans $\sum_{b:\mathcal{D}_t^{\oplus}} a \cong_t^{\oplus} b$ (dually, $\sum_{b:\mathcal{D}_l^{\ominus}} a \cong_l^{\ominus} b$) by Lemma 4.11. The latter type is a proposition by assumption. □

For split preduploids, Definition 4.15 is incorrect in general: the polarity mapping gives a way to differentiate between syntactically positive and negative objects.

**Lemma 4.18** (neg_is_duploid_univalent_if_split)**.** *Let $\mathcal{D}$ be a split duploid. If there is an object $a : \mathcal{D}$ which is both semantically positive and semantically negative, then the underlying single-sorted duploid of $\mathcal{D}$ is not univalent.*

*Proof.* If $a$ is both positive and negative, then we have (by Lemma 4.12) the linear-and-thunkable isomorphism

$$\Uparrow a \overset{\mathtt{force}_a}{\cong_{lt}} a \overset{\mathtt{wrap}_a}{\cong_{lt}} \Downarrow a\,.$$

If $\mathcal{D}$ is univalent, then we have $\Uparrow a =_{\mathsf{ob}\,\mathcal{D}} \Downarrow a$ by the above. Thus, in particular, the polarity mapping would give

$$\ominus = \omega(\Uparrow a) = \omega(\Downarrow a) = \oplus\,,$$

which is a contradiction. Therefore $\mathcal{D}$ is not univalent. □

**Definition 4.19** (is_split_duploid_univalent)**.** A split duploid or preduploid $\mathcal{D}$ is *univalent* (or *univalent as a split duploid* for emphasis) if both its positive-thunkable subcategory $\mathcal{D}_t^{\oplus}$ and and its negative-linear subcategory $\mathcal{D}_l^{\ominus}$ are univalent.

*Remark* 4.20. Definition 4.19 coincides with the indiscernibilities defined for two-sorted duploids by Ahrens et al. [2].

**Theorem 4.21** (isaprop_has_polarity_shifts)**.** *In a univalent preduploid, the type of "having polarity shifts" becomes a mere proposition. That is, polarity shifts become unique if they exist.*

*Proof.* Suppose there are two negative shifts $\langle\Uparrow, \mathtt{force}\rangle$ and $\langle\Uparrow', \mathtt{force}'\rangle$ for a preduploid $\mathcal{D}$. Then for all objects $a$, there is a linear isomorphism

$$f : \Uparrow a \overset{\mathtt{force}_a}{\cong_l} a \overset{\mathtt{delay}'_a}{\cong_l} \Uparrow' a$$

that is also thunkable by the polarities of $\Uparrow$ and $\Uparrow'$. Moreover, $\mathtt{force}_a : \Uparrow a \to a$ transported across this linear-and-thunkable isomorphism, as $f^{-1} \fatsemi \mathtt{force}_a : \Uparrow' a \to a$, is equal to $\mathtt{force}'_a : \Uparrow' a \to a$:

$$\begin{aligned} f^{-1} \fatsemi \mathtt{force}_a &= (\mathtt{force}_a \fatsemi \mathtt{delay}'_a)^{-1} \fatsemi \mathtt{force}_a \\ &= \mathtt{force}'_a \fatsemi \mathtt{delay}_a \fatsemi \mathtt{force}_a \\ &= \mathtt{force}'_a\,. \end{aligned}$$

It follows from univalence of $\mathcal{D}$ that $\langle\Uparrow, \mathtt{force}\rangle = \langle\Uparrow', \mathtt{force}'\rangle$. The case is analogous for positive shifts. □

# 5. ROUGH The Duploids arising from an Adjunction

## 5.1. ROUGH The Oblique Duploid

**Definition 5.1** (adjunction [1])**.** An *adjunction* between categories $\mathcal{P}$ and $\mathcal{N}$ consists of two functors $L\colon \mathcal{P} \to \mathcal{N}$ and $R\colon \mathcal{N} \to \mathcal{P}$, along with two natural transformations: the *unit* $\eta\colon \mathsf{Id}_{\mathcal{P}} \Rightarrow RL$ and *counit* $\epsilon\colon LR \Rightarrow \mathsf{Id}_{\mathcal{N}}$. These must satisfy the "triangle equations":

$$\begin{aligned} L\eta_p \fatsemi \epsilon_{Fp} &= \mathsf{id}_{Fp} && \text{for all } p : \mathcal{P}, \text{ and} \\ \eta_{Gn} \fatsemi G\epsilon_n &= \mathsf{id}_{Gn} && \text{for all } n : \mathcal{N}. \end{aligned}$$

The adjunction itself may be written as any variant of

$$L \dashv R\,, \qquad L \dashv_{(\eta,\epsilon)} R\,, \qquad L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}\,, \quad \text{or} \quad \begin{tikzcd} \mathcal{P} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{N} \arrow[l, bend left, "R"] \end{tikzcd}\ .$$

*Remark* 5.2 (natural_hom_weq, natural_bi_hom_weq). It is well-known that another way to formulate an adjunction $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ is as an isomorphism of sets:

$$\mathcal{N}[\![Lp, n]\!] \overset{\varphi_{p,n}}{\simeq} \mathcal{P}[\![p, Rn]\!] \qquad \text{natural in } p : \mathcal{P}^{\mathsf{op}} \text{ and } n : \mathcal{N}. \tag{5.1}$$

Explicitly, $\varphi$ may be given in terms of $\eta$ and $\epsilon$ as

$$\begin{aligned} \varphi_{p,n}\left(Lp \xrightarrow{f} n\right) &\overset{\text{def}}{\equiv} p \xrightarrow{\eta_p} RLp \xrightarrow{Rf} Rn \\ \varphi^{-1}_{p,n}\left(p \xrightarrow{g} Rn\right) &\overset{\text{def}}{\equiv} Lp \xrightarrow{Lg} LRn \xrightarrow{\epsilon_n} n\,. \end{aligned}$$

Following Levy [15], we may call the morphisms on either side of $\varphi$ (5.1) the *oblique morphisms*, adopting the following definition.

**Definition 5.3** (oblique_mor)**.** Let $\varphi_{p,n} : \mathcal{N}[\![Lp, n]\!] \simeq \mathcal{P}[\![p, Rn]\!]$ be the natural isomorphism witnessing an adjunction $L \dashv R$. An *oblique morphism* $f$ in $L \dashv R$ from $p : \mathcal{P}$ to $n : \mathcal{N}$, written $f : \mathcal{O}_{L \dashv R}[\![p, n]\!]$ or simply $f : \mathcal{O}[\![p, n]\!]$, is a pair of *mates*: a morphism $f^\flat : \mathcal{N}[\![Lp, n]\!]$ and a morphism $f^\sharp : \mathcal{P}[\![p, Rn]\!]$ such that $\varphi_{p,n}(f^\flat) = f^\sharp$.

*Convention* 5.4. We will draw diagrams involving oblique morphisms in two columns, one on the left for the $\mathcal{N}$ mate and another on the right for the $\mathcal{P}$ mate. For example, $f : \mathcal{O}[\![p, n]\!]$ on its own may be depicted as

$$Lp \xrightarrow{f^\flat} n \qquad\qquad p \xrightarrow{f^\sharp} Rn \quad .$$

*Remark* 5.5 (oblique_compose_negative, ...). $\mathcal{O}$ extends to a functor $\mathcal{P}^{\mathsf{op}} \times \mathcal{N} \to \mathbf{Set}$ by pre- and post-composition in the respective categories, as

$$\begin{tikzcd}
Lp \arrow[d, "Lf"'] & & & p \arrow[d, "f"'] & \\
Lq \arrow[r, "g^\flat"] & n \arrow[d, "h"] & & q \arrow[r, "g^\sharp"] & Rn \arrow[d, "Rh"] \\
& m & & & Rm \, ,
\end{tikzcd}$$

with the two sides being mates precisely by naturality of $\varphi$.

*Remark* 5.6 (isweq_oblique_mor_negative, ...). The equation $\varphi_{p,n}(f^\flat) = f^\sharp$ for an oblique morphism $f : \mathcal{O}[\![p, n]\!]$ means that each mate is uniquely determined by the other. The projections $\flat$ and $\sharp$ thus have the obvious inverses,

$$\flat^{-1} : \mathcal{N}[\![Lp, n]\!] \to \mathcal{O}[\![p, n]\!] \quad \text{and} \quad \sharp^{-1} : \mathcal{P}[\![p, Rn]\!] \to \mathcal{O}[\![p, n]\!] \, ,$$

determined by

$$\begin{aligned}
\left(g^{\flat^{-1}}\right)^\sharp &\stackrel{\text{def}}{\equiv} \varphi_{p,n}(g) : \mathcal{P}[\![n, Rp]\!] && \text{for } g : \mathcal{N}[\![Ln, p]\!], \text{ and} \\
\left(h^{\sharp^{-1}}\right)^\flat &\stackrel{\text{def}}{\equiv} \varphi_{p,n}^{-1}(h) : \mathcal{N}[\![Lp, n]\!] && \text{for } h : \mathcal{P}[\![n, Rp]\!].
\end{aligned}$$

This way we have

$$\mathcal{N}[\![Lp, n]\!] \overset{\flat^{-1}}{\simeq} \mathcal{O}[\![p, n]\!] \overset{\sharp}{\simeq} \mathcal{P}[\![p, Rn]\!] \qquad \text{natural in } p : \mathcal{P}^{\mathsf{op}} \text{ and } n : \mathcal{N}.$$

*Remark* 5.7. We use the set of oblique morphisms $\mathcal{O}[\![p, n]\!]$ rather than taking an arbitrary bias towards either $\mathcal{N}[\![Lp, n]\!]$ or $\mathcal{P}[\![p, Rn]\!]$. Logically, this does not matter due to Remark 5.6. However, it has the practical benefit of making all definitions and proofs symmetric. In particular, both the projections $f^\sharp$ and $f^\flat$ for an oblique morphism $f : \mathcal{O}[\![p, n]\!]$ may provided as judgemental equalities—so long as they are proven to be mates—allowing for easier mechanization in a proof assistant.

NOTE: The oblique duploid formalization considerably diverges from this description. The formalization defines it more directly as Guillaume described it, but ported to oblique morphisms, rather than taking a bias. The description of composition and shifts here is actually a description of the envelope duploid.

**Definition 5.8** (`oblique_ob`, `oblique_mor'`)**.** Let $L \dashv R : \mathcal{N} \to \mathcal{P}$ be an adjunction. The *oblique duploid-to-be* $\mathcal{O}[L \dashv R]$ over $L \dashv R$ has objects and morphisms defined as follows.

- The objects $\mathsf{ob}\,\mathcal{O}[L \dashv R] \overset{\text{def}}{\equiv} \mathcal{N} \amalg \mathcal{P}$ are the objects from either $\mathcal{N}$ or $\mathcal{P}$. These are written respectively $\iota^\ominus_{\mathcal{O}}\, n$ for $n : \mathcal{N}$, and $\iota^\oplus_{\mathcal{O}}\, p$ for $p : \mathcal{P}$.
- For an object $a : \mathsf{ob}\,\mathcal{O}[L \dashv R]$, define the *negative* and *positive projections* $\pi^\ominus_{\mathcal{O}}\, a : \mathcal{N}$ and $\pi^\oplus_{\mathcal{O}}\, a : \mathcal{P}$ as

$$\pi^\ominus_{\mathcal{O}}(\iota^\ominus\, n) \overset{\text{def}}{\equiv} n \qquad\qquad \pi^\oplus_{\mathcal{O}}(\iota^\ominus\, n) \overset{\text{def}}{\equiv} Ln$$
$$\pi^\ominus_{\mathcal{O}}(\iota^\oplus\, p) \overset{\text{def}}{\equiv} Rp \qquad\qquad \pi^\oplus_{\mathcal{O}}(\iota^\oplus\, p) \overset{\text{def}}{\equiv} p\,.$$

- The morphisms $\mathcal{O}[L \dashv R][\![a, b]\!]$ are given by the oblique morphisms $\mathcal{O}_{L\dashv R}[\![\pi^\oplus\, a, \pi^\ominus\, b]\!]$ between the positive and negative projections.

**Definition 5.9** (`oblique_negative_identity`, ...)**.** For an object $a : \mathcal{O}[L \dashv R]$ of the oblique duploid-to-be, its *cross morphism* $\mathsf{cross}_{\mathcal{O}}\, a$ is an oblique morphism $\mathsf{cross}_{\mathcal{O}}\, a : \mathcal{O}_{L\dashv R}[\![\pi^\oplus\, a, \pi^\ominus\, a]\!]$, defined as

$$\mathsf{cross}_{\mathcal{O}}(\iota^\ominus\, n)^\flat \overset{\text{def}}{\equiv} \epsilon_n \qquad\qquad \mathsf{cross}_{\mathcal{O}}(\iota^\oplus\, p)^\flat \overset{\text{def}}{\equiv} \mathsf{id}_{Lp}$$
$$\mathsf{cross}_{\mathcal{O}}(\iota^\ominus\, n)^\sharp \overset{\text{def}}{\equiv} \mathsf{id}_{Rn} \qquad\qquad \mathsf{cross}_{\mathcal{O}}(\iota^\oplus\, p)^\sharp \overset{\text{def}}{\equiv} \eta_p\,.$$

That is, the cross morphism $\mathsf{cross}_{\mathcal{O}}\, a$ has components

$$\begin{array}{lll} LRn \xrightarrow{\epsilon_n} n & Rn \xrightarrow{\mathsf{id}_{Rn}} Rn & \text{when } a = \iota^{\ominus}_{\mathcal{O}}\, p, \text{ and} \\ Lp \xrightarrow{\mathsf{id}_{Lp}} Lp & p \xrightarrow{\eta_p} RLp & \text{when } a = \iota^{\oplus}_{\mathcal{O}}\, p\,. \end{array}$$

*Notation* 5.10. We shall freely omit the subscript $\mathcal{O}$ from $\iota^{\oplus}_{\mathcal{O}}$, $\iota^{\ominus}_{\mathcal{O}}$, $\pi^{\oplus}_{\mathcal{O}}$, $\pi^{\ominus}_{\mathcal{O}}$, and $\mathsf{cross}_{\mathcal{O}}$, particularly when they should be considered opaque.

*Notation* 5.11. The operator $\mathsf{cross}$ should be read with a higher precedence than $\flat$ and $\sharp$. That is, the following all mean the same thing:

$$(\mathsf{cross}\, a)^{\flat} \qquad \mathsf{cross}(a)^{\flat} \qquad \mathsf{cross}\, a^{\flat}\,.$$

**Definition 5.12** (`oblique_identity`)**.** The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ has identities given by the cross morphisms. That is, for $a : \mathcal{O}[L \dashv R]$ define $\mathsf{id}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}\, a$ of type $\mathcal{O}[L \dashv R][\![a, a]\!] \equiv \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!]$.

How should we compose the morphisms of the oblique duploid-to-be? Given morphisms $f : \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, b]\!]$ and $g : \mathcal{O}[\![\pi^{\oplus}\, b, \pi^{\ominus}\, c]\!]$, we must find an oblique morphism $f \fatsemi g : \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, c]\!]$. Since composition shall need to be unital, it must satisfy $\mathsf{cross}\, b \fatsemi g = g$ and $f \fatsemi \mathsf{cross}\, b = f$. Let us therefore depict $f$, $g$, and $\mathsf{cross}\, b$ on a diagram:

$$\begin{tikzcd} L(\pi^{\oplus}\, a) \arrow[dr, "f^{\flat}"] & \\ L(\pi^{\oplus}\, b) \arrow[r, dashed, "\mathsf{cross}\, b^{\flat}"] \arrow[dr, "g^{\flat}"'] & \pi^{\ominus}\, b \\ & \pi^{\ominus}\, c \end{tikzcd} \qquad \begin{tikzcd} \pi^{\oplus}\, a \arrow[dr, "f^{\sharp}"] & \\ \pi^{\oplus}\, b \arrow[r, dashed, "\mathsf{cross}\, b^{\sharp}"] \arrow[dr, "g^{\sharp}"'] & R(\pi^{\ominus}\, b) \\ & R(\pi^{\ominus}\, c) \end{tikzcd}$$

We need to find a composite in either category: either $(f \fatsemi g)^{\flat} : \mathcal{N}[\![L(\pi^{\oplus}\, a), \pi^{\ominus}\, c]\!]$ or $(f \fatsemi g)^{\sharp} : \mathcal{P}[\![\pi^{\oplus}\, a, R(\pi^{\ominus}\, c)]\!]$. If we had a morphism in the opposite direction to $\mathsf{cross}\, b$ in either category, so one $h : \mathcal{N}[\![\pi^{\ominus}\, b, L(\pi^{\oplus}\, b)]\!]$ or $k : \mathcal{P}[\![R(\pi^{\ominus}\, b), \pi^{\oplus}\, b]\!]$, then we could simply compose it between $f$ and $g$ as $f^{\flat} \fatsemi h \fatsemi g^{\flat}$ or $f^{\sharp} \fatsemi k \fatsemi g^{\sharp}$ respectively. We do have such a morphism: recall from Definition 5.9 that either $\mathsf{cross}_{\mathcal{O}}\, b^{\flat}$ or $\mathsf{cross}_{\mathcal{O}}\, b^{\sharp}$ is the identity, and thus an isomorphism.

**Definition 5.13** (`oblique_compose`). The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ has composition defined as follows. Let $f : \mathcal{O}[L \dashv R][\![a, b]\!]$ and $g : \mathcal{O}[L \dashv R][\![b, c]\!]$ be morphisms in the oblique duploid. Define $f \fatsemi g : \mathcal{O}[L \dashv R][\![a, c]\!]$ by discrimination on $b$, as either

$$
\begin{aligned}
(f \fatsemi g)^\flat &\overset{\text{def}}{\equiv} L(\pi^\oplus a) \xrightarrow{Lf^\sharp} LR(\pi^\ominus b) \xrightarrow{L(\mathsf{cross}_{\mathcal{O}}\, b^\sharp)^{-1}} L(\pi^\oplus b) \xrightarrow{g^\flat} \pi^\ominus c && \text{and} \\
(f \fatsemi g)^\sharp &\overset{\text{def}}{\equiv} \pi^\oplus a \xrightarrow{f^\sharp} R(\pi^\ominus b) \xrightarrow{(\mathsf{cross}_{\mathcal{O}}\, b^\sharp)^{-1}} \pi^\oplus b \xrightarrow{g^\sharp} R(\pi^\ominus c) && \text{if } b \equiv \iota^\ominus n\text{, or} \\
(f \fatsemi g)^\flat &\overset{\text{def}}{\equiv} L(\pi^\oplus a) \xrightarrow{f^\flat} \pi^\ominus b \xrightarrow{(\mathsf{cross}_{\mathcal{O}}\, b^\flat)^{-1}} L(\pi^\oplus b) \xrightarrow{g^\flat} \pi^\ominus c && \text{and} \\
(f \fatsemi g)^\sharp &\overset{\text{def}}{\equiv} \pi^\oplus a \xrightarrow{f^\sharp} R(\pi^\ominus b) \xrightarrow{R(\mathsf{cross}_{\mathcal{O}}\, b^\flat)^{-1}} RL(\pi^\oplus b) \xrightarrow{Rg^\flat} R(\pi^\ominus c) && \text{if } b \equiv \iota^\oplus p\,.
\end{aligned}
$$

*Remark* 5.14. Definition 5.13 "forgets" that the mentioned isomorphisms $\mathsf{cross}_{\mathcal{O}}\, b^\sharp$ and $\mathsf{cross}_{\mathcal{O}}\, b^\flat$ are the identity, in anticipation of the generalization in Section 5.2. If we remember this fact, we can instead obtain, for $a \xrightarrow{f} b \xrightarrow{g} c$, the definitions

$$
\begin{aligned}
(f \fatsemi g)^\sharp &\overset{\text{def}}{\equiv} \pi^\oplus a \xrightarrow{f^\sharp} Rn \xrightarrow{g^\sharp} R(\pi^\ominus c) && \text{if } b \equiv \iota^\ominus n \\
(f \fatsemi g)^\flat &\overset{\text{def}}{\equiv} L(\pi^\oplus a) \xrightarrow{f^\flat} Lp \xrightarrow{g^\flat} \pi^\ominus c && \text{if } b \equiv \iota^\oplus p\,.
\end{aligned}
$$

This is the actual definition of `oblique_compose` used in the formalization, and by the original construction of Munch-Maccagnoni [23].

**Lemma 5.15** (`oblique_left_id`, …). *The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ with composition as in Definition 5.13 is unital (Definition 3.1.4.a), and thus a unital magmoid.*

*Proof.* Let $f : a \to b$ in $\mathcal{O}[L \dashv R]$. Let us show that $\mathsf{id}_a \fatsemi f = f$. Suppose $a \equiv \iota^\oplus p$; then we have

$$
\begin{aligned}
(\mathsf{cross}\, a \fatsemi f)^\flat &\equiv \mathsf{cross}\, a^\flat \fatsemi (\mathsf{cross}\, a^\flat)^{-1} \fatsemi f^\flat \\
&= f^\flat\,.
\end{aligned}
$$

Thus we have $\mathsf{cross}\, a \fatsemi f = f$ by injectivity of the equivalence $\flat$. The cases for $a \equiv \iota^\ominus n$ and the two cases for post-composition are all analogous. □

**Lemma 5.16** (`oblique_negative_is_negative`, …). *In the oblique duploid-to-be $\mathcal{O}[L \dashv R]$, the objects $\iota^\ominus n$ for $n : \mathcal{N}$ are semantically negative, and those $\iota^\oplus p$ for $p : \mathcal{P}$ are semantically positive (Definition 3.19).*

*Proof (sketch).* To see that all objects $\iota^\ominus n$ are negative—that is, all triples $a \xrightarrow{f} x \xrightarrow{g} c \xrightarrow{h} d$ with $x \equiv \iota^\ominus n$ associate—calculate $(f \fatsemi (g \fatsemi h))^\sharp = ((f \fatsemi g) \fatsemi h)^\sharp$ by discriminating on $c$.[1] The other polarity is symmetric. □

**Corollary 5.17** (oblique_preduploid)**.** *The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ is a split preduploid with polarity mapping $\omega\colon \mathsf{ob}\,\mathcal{O}[L \dashv R] \to \{\ominus, \oplus\}$ defined as*

$$\omega(\iota^{\ominus}\, n) \overset{\text{def}}{\equiv} \ominus$$
$$\omega(\iota^{\oplus}\, p) \overset{\text{def}}{\equiv} \oplus\,.$$

**Definition 5.18** (oblique_negative_shift_data, …)**.** The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ has polarity shift structure (Definition 4.2) given on objects by

$$\Uparrow a \overset{\text{def}}{\equiv} \iota^{\ominus}(\pi^{\ominus}\, a) \quad \text{and} \quad \Downarrow a \overset{\text{def}}{\equiv} \iota^{\oplus}(\pi^{\oplus}\, a)\,.$$

For the morphism $\mathtt{force}_a : \mathcal{O}[L \dashv R][\![\Uparrow a, a]\!]$, compute its type

$$\begin{aligned} \mathcal{O}[L \dashv R][\![\Uparrow a, a]\!] &\equiv \mathcal{O}[\![\pi^{\oplus}(\Uparrow a), \pi^{\ominus}\, a]\!] \\ &\equiv \mathcal{O}[\![\pi^{\oplus}(\Uparrow a), \pi^{\ominus}(\Uparrow a)]\!]\,, \end{aligned}$$

and give $\mathtt{force}_a \overset{\text{def}}{\equiv} \mathsf{cross}\Uparrow a : \mathcal{O}[\![\pi^{\oplus}(\Uparrow a), \pi^{\ominus}(\Uparrow a)]\!]$. For its inverse-to-be $\mathtt{delay}_a$ : $\mathcal{O}[L \dashv R][\![a, \Uparrow a]\!]$, compute

$$\begin{aligned} \mathcal{O}[L \dashv R][\![a, \Uparrow a]\!] &\equiv \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus} \Uparrow a]\!] \\ &\equiv \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!] \end{aligned}$$

and give $\mathtt{delay}_a \overset{\text{def}}{\equiv} \mathsf{cross}\, a : \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!]$.

Dually, define $\mathtt{wrap}_a \overset{\text{def}}{\equiv} \mathsf{cross}\Downarrow a$ of the type $\mathcal{O}[L \dashv R][\![a, \Downarrow a]\!] \equiv \mathcal{O}[\![\pi^{\oplus}(\Downarrow a), \pi^{\ominus}(\Downarrow a)]\!]$, and $\mathtt{unwrap}_a \overset{\text{def}}{\equiv} \mathsf{cross}\, a$ of the type $\mathcal{O}[L \dashv R][\![\Downarrow a, a]\!] \equiv \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!]$.

**Lemma 5.19** (oblique_negative_shift_axioms, …)**.** *The polarity shift structure of Definition 5.18 for the oblique duploid-to-be $\mathcal{O}[L \dashv R]$ satisfies the axioms of polarity shifts (Definition 4.2), and respects the polarity mapping.*

*Proof (sketch).* The polarities of the shifts follow from Lemma 5.16. We leave it to the reader to check the remainder of the axioms. Alternatively, see Proposition II.20 of Munch-Maccagnoni, PhD thesis. □

[1]Bonus points for forgetting $\mathsf{cross}\, x^{\sharp} \equiv \mathsf{id}_{Rn}$, beyond the existence of $(\mathsf{cross}\, x^{\sharp})^{-1}$.

**Corollary 5.20** (oblique_duploid, oblique_split_duploid)**.** *The oblique duploid-to-be $\mathcal{O}[L \dashv R]$ is a split duploid. Hence, we now call it the* oblique duploid*.*

Similarly to Lemma 4.10, we have the following characterizations of linearity, thunkabilitiy and polarities in the oblique duploid.

**Lemma 5.21** (is_thunkable_iff_oblique_unit_postcompose, ... [23])**.** *Let $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ be an adjunction. A morphism $f : a \to \iota^{\oplus}_{\mathcal{O}}\, p$ in the oblique duploid $\mathcal{O}[L \dashv R]$ is thunkable if and only if the equation*

$$f^{\sharp} \fatsemi RL\eta_p = f^{\sharp} \fatsemi \eta_{RLp} \quad \textit{holds.} \tag{5.2}$$

*Dually, a morphism $g : \iota^{\ominus}_{\mathcal{O}}\, n \to b$ in the oblique duploid $\mathcal{O}[L \dashv R]$ is linear if and only if the equation*

$$LR\epsilon_n \fatsemi g^{\flat} = \epsilon_{RLn} \fatsemi g^{\flat} \quad \textit{holds.}$$

*Proof.* We show only the thunkability case, with the linearity following by duality. By Lemma 4.10, thunkability of $f$ is equivalent to the associativity of the triple

$$a \xrightarrow{f} \iota^{\oplus}_{\mathcal{O}}\, p \xrightarrow{\mathtt{delay}_{\iota^{\oplus}_{\mathcal{O}} p}} \Uparrow(\iota^{\oplus}_{\mathcal{O}}\, p) \xrightarrow{\mathtt{wrap}_{\Uparrow\iota^{\oplus}_{\mathcal{O}} p}} \Downarrow\Uparrow(\iota^{\oplus}_{\mathcal{O}}\, p)\,. \tag{5.3}$$

We may calculate

$$\begin{aligned} (f \fatsemi (\mathtt{delay}_{\iota^{\oplus}_{\mathcal{O}} p} \fatsemi \Uparrow(\iota^{\oplus}_{\mathcal{O}}\, p)))^{\sharp} &= f^{\sharp} \fatsemi RL\eta_p && \text{on the left, and} \\ ((f \fatsemi \mathtt{delay}_{\iota^{\oplus}_{\mathcal{O}} p}) \fatsemi \Uparrow(\iota^{\oplus}_{\mathcal{O}}\, p))^{\sharp} &= f^{\sharp} \fatsemi \eta_{RLp} && \text{on the right.} \end{aligned}$$

Thus, associativity of Equation 5.3 is equivalent to Equation 5.2. □

**Lemma 5.22** (is_negative_oblique_positive_iff_pre_fixed_point, ...)**.** *Let $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ be an adjunction. For all $p : P$, the object $\iota^{\oplus}_{\mathcal{O}}\, p$ is negative in $\mathcal{O}[L \dashv R]$ if and only if the equation*

$$RL\eta_p = \eta_{RLp} \quad \textit{holds.}$$

*Dually, for $n : N$, the object $\iota^{\ominus}_{\mathcal{O}}\, n$ is positive in $\mathcal{O}[L \dashv R]$ if and only if the equation*

$$LR\epsilon_n = \epsilon_{LRn} \quad \textit{holds.}$$

*Proof.* Combine Lemma 4.12 and Lemma 5.21. □

It follows from Lemma 4.18 that the oblique duploid is not in general univalent.

**Example 5.23.** Let $\mathcal{C}$ be a category with at least one object $a : \mathcal{C}$. Then the object $\iota^{\oplus}_{\mathcal{O}}\, a$ is both semantically negative and positive in the oblique duploid over the identity adjunction $\mathcal{O}[\mathsf{Id}_{\mathcal{C}} \vdash \mathsf{Id}_{\mathcal{C}}]$. Therefore the latter is not univalent as a single-sorted duploid.

## 5.2. ROUGH The Envelope Duploid

As anticipated in Remark 5.14, we obtain a perfectly serviceable version of the oblique duploid by treating the cross morphisms as merely isomorphisms.

**Definition 5.24** (envelope_preob)**.** Let $L \dashv R : \mathcal{N} \to \mathcal{P}$ be an adjunction. An *envelope pre-object* in $L \dashv R$ is a triple of a positive object $p : \mathcal{P}$, a negative object $n : \mathcal{N}$, and an oblique morphism $f : \mathcal{O}_{L\dashv R}[\![p, n]\!]$ between them.

*Notation* 5.25. For an envelope pre-object $a$ in $L \dashv R$, write $\pi^{\oplus}_{\mathcal{E}}\, a : \mathcal{P}$ for its positive object, $\pi^{\ominus}_{\mathcal{E}}\, a : \mathcal{N}$ for its negative object, and $\mathsf{cross}_{\mathcal{E}}\, a : \mathcal{O}_{L\dashv R}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!]$ for its oblique morphism.

*Notation* 5.26. We may write any envelope pre-object $a$ as simply the diagram of its oblique morphism:

$$\pi^{\oplus}_{\mathcal{E}}\, a \xrightarrow{\mathsf{cross}_{\mathcal{E}}\, a} \pi^{\ominus}_{\mathcal{E}}\, a\,.$$

**Definition 5.27** (envelope_preob_of_negative, ...)**.** For an object $a : \mathcal{O}[L \dashv R]$ of the oblique duploid, define its corresponding envelope pre-object $\hat{a}$ as

$$\left(\pi^{\oplus}_{\mathcal{E}}\, \hat{a} \xrightarrow{\mathsf{cross}_{\mathcal{E}}\, \hat{a}} \pi^{\ominus}_{\mathcal{E}}\, \hat{a}\right) \overset{\text{def}}{\equiv} \left(\pi^{\oplus}_{\mathcal{O}}\, a \xrightarrow{\mathsf{cross}_{\mathcal{O}}\, a} \pi^{\ominus}_{\mathcal{O}}\, a\right)\,.$$

Define the negative and positive injections for $n : \mathcal{N}$ or $p : \mathcal{P}$ as

$$\iota^{\ominus}_{\mathcal{E}}\, n \overset{\text{def}}{\equiv} \widehat{\iota^{\ominus}_{\mathcal{O}}\, n} \quad \text{and} \quad \iota^{\oplus}_{\mathcal{E}}\, p \overset{\text{def}}{\equiv} \widehat{\iota^{\oplus}_{\mathcal{O}}\, p}\,.$$

*Notation* 5.28. We shall freely omit the subscript $\mathcal{E}$ from $\iota^{\oplus}_{\mathcal{E}}$, $\iota^{\ominus}_{\mathcal{E}}$, $\pi^{\oplus}_{\mathcal{E}}$, $\pi^{\ominus}_{\mathcal{E}}$, and $\mathsf{cross}_{\mathcal{E}}$. The conflation with the $\mathcal{O}$-subscripted variants is intentional.

**Definition 5.29** (envelope_polarization_choice)**.** Let $a$ be an envelope pre-object in $L \dashv R$. A *polarization choice* for $a$ is a (two-sided) inverse for either mate of

$\mathsf{cross}\, a : \mathcal{O}_{L\dashv R}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, a]\!]$. That is, either

$$\begin{array}{lll} \text{an inverse} & R(\pi^{\ominus}\, a) \xrightarrow{(\mathsf{cross}\, a^{\sharp})^{-1}} \pi^{\oplus}\, a & \text{for } \mathsf{cross}\, a^{\sharp}\text{, or} \\ \text{an inverse} & \pi^{\ominus}\, a \xrightarrow{(\mathsf{cross}\, a^{\flat})^{-1}} L(\pi^{\oplus}\, a) & \text{for } \mathsf{cross}\, a^{\flat}\,. \end{array}$$

We call the former a *negative polarization choice*, and the latter a *positive polarization choice*.

**Lemma 5.30** (envelope_chosen_negative_of_negative, …)**.** *For an object $a : \mathcal{O}[L \dashv R]$ of the oblique duploid, its corresponding envelope pre-object $\hat{a}$ comes equipped with a corresponding polarization choice, negative if $a \equiv \iota^{\ominus}_{\mathcal{O}}\, n$ and positive if $a \equiv \iota^{\oplus}_{\mathcal{O}}\, p$.*

*Proof.* This amounts to the computations $\mathsf{cross}_{\mathcal{O}}(\iota^{\ominus}_{\mathcal{O}}\, n)^{\sharp} \equiv \mathsf{id}_{Rn}$ and $\mathsf{cross}_{\mathcal{O}}(\iota^{\oplus}_{\mathcal{O}}\, p)^{\flat} \equiv \mathsf{id}_{Lp}$. □

**Lemma 5.31** (weq_oblique_ob_to_split_envelope_ob)**.** *Let $L \dashv R : \mathcal{N} \to \mathcal{P}$ be an adjunction between **univalent** categories $\mathcal{N}$ and $\mathcal{P}$. The mapping of an oblique duploid object $a : \mathcal{O}[L \dashv R]$ to its corresponding envelope pre-object $\hat{a}$ equipped with a polarization choice is an equivalence.*

*Proof (sketch).* The inverse is given by $\pi^{\ominus}_{\mathcal{E}}$ or $\pi^{\oplus}_{\mathcal{E}}$ depending on the polarization choice—negative or positive respectively—of the envelope pre-object. The proof that this is an inverse is omitted. □

**Definition 5.32** (envelope_ob, envelope_mor)**.** The *envelope duploid-to-be* $\mathcal{E}[L \dashv R]$ has objects, morphisms, and identities given as follows.

- The objects $\mathsf{ob}\, \mathcal{E}[L \dashv R]$ are envelope pre-objects $a$ in $L \dashv R$ such that there exists a polarization choice for $a$. That is, $\mathsf{cross}\, a^{\sharp}$ or $\mathsf{cross}\, a^{\flat}$ is an isomorphism.
- Morphisms $\mathcal{E}[L \dashv R][\![a, b]\!]$ are the oblique morphisms $\mathcal{O}_{L\dashv R}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, b]\!]$.
- For an object $a : \mathcal{E}[L \dashv R]$, its identity is given by its cross morphism $\mathsf{id}_a \overset{\text{def}}{\equiv} \mathsf{cross}\, a$.

*Remark* 5.33. Beware Convention 2.15: the polarization choice for each envelope object is under propositional truncation. Importantly, it does not partake in identifications of envelope objects. However, it may not be obtained constructively unless the choice is irrelevant, such as when proving another proposition.

**Definition 5.34** (envelope_compose'). Let $a$, $b$, and $c$ be envelope pre-objects in $L \dashv R$, and let $f : \mathcal{O}[\![\pi^{\oplus}\, a, \pi^{\ominus}\, b]\!]$ and $g : \mathcal{O}[\![\pi^{\oplus}\, b, \pi^{\ominus}\, c]\!]$ be oblique morphisms between them. Given a polarization choice for $b$, define depending on it

$$
\begin{aligned}
(f \mathbin{⨾} g)^{\flat} &\overset{\text{def}}{\equiv} L(\pi^{\oplus}\, a) \xrightarrow{Lf^{\sharp}} LR(\pi^{\ominus}\, b) \xrightarrow{L(\mathsf{cross}_{\mathcal{O}}\, b^{\sharp})^{-1}} L(\pi^{\oplus}\, b) \xrightarrow{g^{\flat}} \pi^{\ominus}\, c && \text{and} \\
(f \mathbin{⨾} g)^{\sharp} &\overset{\text{def}}{\equiv} \pi^{\oplus}\, a \xrightarrow{f^{\sharp}} R(\pi^{\ominus}\, b) \xrightarrow{(\mathsf{cross}_{\mathcal{O}}\, b^{\sharp})^{-1}} \pi^{\oplus}\, b \xrightarrow{g^{\sharp}} R(\pi^{\ominus}\, c) && \text{if negative, or}
\end{aligned}
$$

$$
\begin{aligned}
(f \mathbin{⨾} g)^{\flat} &\overset{\text{def}}{\equiv} L(\pi^{\oplus}\, a) \xrightarrow{f^{\flat}} \pi^{\ominus}\, b \xrightarrow{(\mathsf{cross}_{\mathcal{O}}\, b^{\flat})^{-1}} L(\pi^{\oplus}\, b) \xrightarrow{g^{\flat}} \pi^{\ominus}\, c && \text{and} \\
(f \mathbin{⨾} g)^{\sharp} &\overset{\text{def}}{\equiv} \pi^{\oplus}\, a \xrightarrow{f^{\sharp}} R(\pi^{\ominus}\, b) \xrightarrow{R(\mathsf{cross}_{\mathcal{O}}\, b^{\flat})^{-1}} RL(\pi^{\oplus}\, b) \xrightarrow{Rg^{\flat}} R(\pi^{\ominus}\, c) && \text{if positive.}
\end{aligned}
$$

*Remark* 5.35. This is identical to Definition 5.13 of composition in the oblique duploid, but on envelope pre-objects instead.

**Theorem 5.36** (envelope_compose_irrel). *Let $f : \mathcal{E}[L \dashv R][\![a, b]\!]$ and $g : \mathcal{E}[L \dashv R][\![b, c]\!]$ be morphisms in the envelope duploid-to-be, such that both negative and positive polarization choices of $b$ exist. Then the two ways of composing $f \mathbin{⨾} g : \mathcal{E}[L \dashv R][\![a, c]\!]$ using Definition 5.34 are equal.*

*Proof.* Let $\varphi_{p,n} : \mathcal{N}[\![Lp, n]\!] \cong \mathcal{P}[\![p, Rn]\!]$ be the natural isomorphism witnessing the adjunction $L \dashv R$. We show that the following morphisms are mates:

$$
\begin{aligned}
&L(\pi^{\oplus}\, a) \xrightarrow{Lf^{\sharp}} LR(\pi^{\ominus}\, b) \xrightarrow{L(\mathsf{cross}\, b^{\sharp})^{-1}} L(\pi^{\oplus}\, b) \xrightarrow{g^{\flat}} \pi^{\ominus}\, c && \text{(the negative choice's } (f \mathbin{⨾} g)^{\flat}) \\
&\pi^{\oplus}\, a \xrightarrow{f^{\sharp}} R(\pi^{\ominus}\, b) \xrightarrow{R(\mathsf{cross}\, b^{\flat})^{-1}} RL(\pi^{\oplus}\, b) \xrightarrow{Rg^{\flat}} R(\pi^{\ominus}\, c) && \text{(the positive choice's } (f \mathbin{⨾} g)^{\sharp}) \,.
\end{aligned}
$$

In fact, we need only check that

$$
R(\pi^{\ominus}\, b) \xrightarrow{R(\mathsf{cross}\, b^{\flat})^{-1}} RL(\pi^{\oplus}\, b) \quad \text{and} \quad LR(\pi^{\ominus}\, b) \xrightarrow{L(\mathsf{cross}\, b^{\sharp})^{-1}} L(\pi^{\oplus}\, b)
$$

are mates, since it follows by naturality of $\varphi$ that

$$
\varphi_{(\pi^{\oplus}\, a),(\pi^{\ominus}\, a)} \left(Lf^{\sharp} \mathbin{⨾} L(\mathsf{cross}\, b^{\sharp})^{-1} \mathbin{⨾} g^{\flat}\right) = \left(f^{\sharp} \mathbin{⨾} R(\mathsf{cross}\, b^{\flat})^{-1} \mathbin{⨾} Rg^{\flat}\right).
$$

Therefore we must show

$$
R(\mathsf{cross}\, b^{\flat})^{-1} = \varphi_{R(\pi^{\ominus}\, b),L(\pi^{\oplus}\, b)}\left(L(\mathsf{cross}\, b^{\sharp})^{-1}\right).
$$

It suffices to consider

$$\begin{aligned}&\varphi^{-1}_{(\pi^\oplus a),L(\pi^\oplus a)}\left(\mathsf{cross}\,b^\sharp \mathbin{⨟} R(\mathsf{cross}\,b^\flat)^{-1}\right)\\&\quad=\varphi^{-1}_{(\pi^\oplus a),L(\pi^\oplus a)}\left(\mathsf{cross}\,b^\sharp \mathbin{⨟} \varphi_{R(\pi^\ominus b),L(\pi^\oplus b)}(L(\mathsf{cross}\,b^\sharp)^{-1})\right)\end{aligned}$$

by isomorphism of $\mathsf{cross}\,b^\sharp$ and $\varphi$. Both sides are equal to $\mathsf{id}_{\pi^\oplus a}$ by naturality of $\varphi$, and isomorphism of $\varphi$ and $\mathsf{cross}\,b^\sharp$. □

**Definition 5.37** (`envelope_compose`). The envelope duploid-to-be $\mathcal{E}[L \dashv R]$ has composition determined by Definition 5.34 from the polarization choice that exists for each envelope object. By Theorem 5.36 the choice of polarization is irrelevant, thus this definition is unique.

**Lemma 5.38** (`envelope_compose_id_left`, ...). *Composition in the envelope duploid-to-be $\mathcal{E}[L \dashv R]$ is unital.*

*Proof.* Same as Lemma 5.15. □

**Lemma 5.39** (`is_negative_of_envelope_chosen_negative`, ...). *Objects $a : \mathcal{E}[L \dashv R]$ in the envelope duploid-to-be with a positive polarization choice are positive, and those with a negative polarization choice are negative.*

*Proof.* Same as Lemma 5.16. □

**Corollary 5.40** (`envelope_preduploid`). *The envelope duploid-to-be $\mathcal{E}[L \dashv R]$ is a single-sorted preduploid.*

**Definition 5.41** (`envelope_negative_shift_data`, ...). The envelope duploid-to-be $\mathcal{E}[L \dashv R]$ has polarity shift structure given the same way as the oblique duploid (Definition 5.18). That is,

$$\begin{aligned}
{\Uparrow} a &\stackrel{\text{def}}{\equiv} \iota^\ominus(\pi^\ominus a) : \mathcal{E}[L \dashv R] & {\Downarrow} a &\stackrel{\text{def}}{\equiv} \iota^\oplus(\pi^\oplus a) : \mathcal{E}[L \dashv R]\\
\mathtt{force}_a &\stackrel{\text{def}}{\equiv} \mathsf{cross}\,{\Uparrow} a : \mathcal{E}[L \dashv R][\![{\Uparrow} a, a]\!] & \mathtt{wrap}_a &\stackrel{\text{def}}{\equiv} \mathsf{cross}\,{\Downarrow} a : \mathcal{E}[L \dashv R][\![a, {\Downarrow} a]\!]\\
\mathtt{delay}_a &\stackrel{\text{def}}{\equiv} \mathsf{cross}\,a : \mathcal{E}[L \dashv R][\![a, {\Uparrow} a]\!] & \mathtt{unwrap}_a &\stackrel{\text{def}}{\equiv} \mathsf{cross}\,a : \mathcal{E}[L \dashv R][\![{\Downarrow} a, a]\!]\,.
\end{aligned}$$

**Lemma 5.42** (`envelope_negative_shift_axioms`, ...). *The polarity shift structure of Definition 5.41 satisfies the axioms of polarity shifts.*

*Proof.* Same as Lemma 5.19. □

**Corollary 5.43** (envelope_duploid)**.** *The envelope duploid-to-be $\mathcal{E}[L \dashv R]$ is a single-sorted duploid. Hence, we now call it the* envelope duploid*.*

The oblique duploid is weakly equivalent to the envelope duploid.

**Lemma 5.44** (weak_equiv_oblique_to_envelope_duploid, …)**.** *The oblique duploid $\mathcal{O}[L \dashv R]$ is weakly equivalent to the corresponding envelope duploid $\mathcal{E}[L \dashv R]$.*

*Proof.* The unital magmoid functor $F : \mathcal{O}[L \dashv R] \to \mathcal{E}[L \dashv R]$ is given on objects by $F_0\, a \overset{\text{def}}{\equiv} \hat{a}$, and is the identity on morphisms $F_1\, f \overset{\text{def}}{\equiv} f$. It is easy to see that this preserves identities and composition in the respective duploids (Definition 5.13, Definition 5.34).

Being the identity on morphisms, $F$ is trivially fully-faithful. It remains to show that it is essentially surjective. For this, we must show given an object $a : \mathcal{E}[L \dashv R]$ that a corresponding object $b : \mathcal{O}[L \dashv R]$ exists such that $\hat{b} \cong_{lt} a$. We shall choose (under propositional truncation) $b \equiv \iota^{\ominus}_{\mathcal{O}}(\pi^{\ominus}_{\mathcal{E}}\, a)$ or $b \equiv \iota^{\oplus}_{\mathcal{O}}(\pi^{\oplus}_{\mathcal{E}}\, a)$ based on the polarization choice of $a$—negative or positive respectively. The linear-and-thunkable isomorphism $\hat{b} \cong_{lt} a$ is then $\Uparrow a \overset{\texttt{force}_a}{\cong_{lt}} a$ or $\Downarrow a \overset{\texttt{unwrap}_a}{\cong_{lt}} a$ depending on the polarization choice. □

*Remark* 5.45 (equiv_oblique_to_envelope_duploid_from_LEM). The equivalence of Lemma 5.44 is strong when assuming the law of excluded middle. This is because we can "decide" whether (say) a negative polarization choice for $a$ exists, and take an arbitrary bias towards $b \equiv \iota^{\ominus}_{\mathcal{O}}(\pi^{\ominus}_{\mathcal{E}}\, a)$ when it does.

**Lemma 5.46** (is_thunkable_from_downshift_iff_envelope_unit_postcompose, … [23])**.** *Let $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ be an adjunction. A morphism $f\colon a \to \iota^{\oplus}_{\mathcal{E}}\, p$ in the envelope duploid $\mathcal{E}[L \dashv R]$ is thunkable if and only if the equation*

$$f^{\sharp} \,⨾\, RL\eta_p = f^{\sharp} \,⨾\, \eta_{RLp} \quad \textit{holds.}$$

*Dually, a morphism $g\colon \iota^{\ominus}_{\mathcal{E}}\, n \to b$ in the envelope duploid $\mathcal{E}[L \dashv R]$ is linear if and only if the equation*

$$LR\epsilon_n \,⨾\, g^{\flat} = \epsilon_{RLn} \,⨾\, g^{\flat} \quad \textit{holds.}$$

*Proof.* Same as Lemma 5.21. □

**Lemma 5.47** (is_negative_envelope_downshift_iff_pre_fixed_point, …)**.** *Let $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ be an adjunction. For all $p : P$, the object $\iota^{\oplus}_{\mathcal{E}}\, p$ is negative in $\mathcal{E}[L \dashv R]$ if and only if the equation*

$$RL\eta_p = \eta_{RLp} \quad \textit{holds.}$$

*Dually, for $n : N$, the object $\iota^{\ominus}_{\mathcal{E}} n$ is positive in $\mathcal{E}[L \dashv R]$ if and only if the equation*

$$LR\epsilon_n = \epsilon_{LRn} \quad \textit{holds.}$$

*Proof.* Combine Lemma 4.12 and Lemma 5.46. □

## 5.3. ROUGH Univalence and the Equalizing Requirement

Under what conditions are the oblique and envelope duploids univalent?

**Lemma 5.48** (`negative_to_oblique_duploid`, `negative_category_to_envelope_duploid`, …). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The inclusions $\iota^{\ominus}_{\mathcal{O}}/\iota^{\oplus}_{\mathcal{O}}$ and $\iota^{\ominus}_{\mathcal{E}}/\iota^{\oplus}_{\mathcal{E}}$ extend to functors mapping $\mathcal{N}/\mathcal{P}$ to their respective syntactically (for $\mathcal{O}$)/semantically (for $\mathcal{E}$) negative-linear and positive-thunkable subcategories:*

$$\iota^{\ominus}_{\mathcal{O}} \colon \mathcal{N} \to \mathcal{O}[L \dashv R]^{\ominus\omega}_{l} \qquad\qquad \iota^{\ominus}_{\mathcal{E}} \colon \mathcal{N} \to \mathcal{E}[L \dashv R]^{\ominus}_{l}$$
$$\iota^{\oplus}_{\mathcal{O}} \colon \mathcal{P} \to \mathcal{O}[L \dashv R]^{\oplus\omega}_{t} \qquad\qquad \iota^{\oplus}_{\mathcal{E}} \colon \mathcal{P} \to \mathcal{E}[L \dashv R]^{\oplus}_{t}\,.$$

*These are given on morphisms by*

$$\iota^{\ominus}\left(n \xrightarrow{g} m\right)^{\flat} \stackrel{\text{def}}{\equiv} LRn \xrightarrow{\epsilon_n \,;\, f} m \qquad\qquad \iota^{\oplus}\left(p \xrightarrow{f} q\right)^{\flat} \stackrel{\text{def}}{\equiv} Lp \xrightarrow{Lf} Lq$$
$$\iota^{\ominus}\left(n \xrightarrow{g} m\right)^{\sharp} \stackrel{\text{def}}{\equiv} Rn \xrightarrow{Rf} Rm \qquad\qquad \iota^{\oplus}\left(p \xrightarrow{f} q\right)^{\sharp} \stackrel{\text{def}}{\equiv} p \xrightarrow{f \,;\, \eta_q} RLq\,.$$

*Proof.* The respective linearity and thunkability conditions follow from Lemma 5.46 and naturality of $\eta/\epsilon$. The functor laws follow from those of $L$ and $R$. □

**Lemma 5.49** (`isweq_on_objects_negative_to_oblique_duploid`, …). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The inclusions $\iota^{\ominus}_{\mathcal{O}} \colon \mathcal{N} \to \mathcal{O}[L \dashv R]^{\ominus\omega}_{l}$ and $\iota^{\oplus}_{\mathcal{O}} \colon \mathcal{P} \to \mathcal{O}[L \dashv R]^{\oplus\omega}_{t}$ are equivalences on objects.*

*Proof.* Immediate from the definition of the polarity mapping of $\mathcal{O}[L \dashv R]$ (Corollary 5.17). The inverses are $\pi^{\ominus}_{\mathcal{O}}$ and $\pi^{\oplus}_{\mathcal{O}}$ respectively. □

**Lemma 5.50** (`envelope_ob_eq_negative'`, …). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction between **univalent** categories $\mathcal{P}$ and $\mathcal{N}$. Every object $a : \mathcal{E}[L \dashv R]$ with a negative polarization choice is identified with $\Uparrow a$. Dually, every object $b : \mathcal{E}[L \dashv R]$ with a positive polarization choice is identified with $\Downarrow b$.*

*Proof.* We show the latter.[2] We need to identify

$$\left(\pi^{\oplus}\, a \xrightarrow{\mathsf{cross}\, a} \pi^{\ominus}\, a\right) = \left(\pi^{\oplus}\, a \xrightarrow{\mathsf{cross}(\Downarrow a)} L(\pi^{\oplus}\, a)\right) . \tag{5.4}$$

We have that $\mathsf{cross}\, a^{\flat} : \mathcal{N}[\![L(\pi^{\oplus}\, a), \pi^{\ominus}\, a]\!]$ is an isomorphism. Thus by Lemma 3.8 we have

$$\langle \pi^{\ominus}\, a, \mathsf{cross}\, a^{\flat} \rangle = \langle L(\pi^{\oplus}\, a), \mathsf{id}_{L(\pi^{\oplus}\, a)} \rangle$$

as elements of the fan $\sum_{n:\mathcal{N}} L(\pi^{\oplus}\, a) \cong n$. Since $\mathsf{cross}(\Downarrow a)^{\flat} \equiv \mathsf{id}_{L(\pi^{\oplus}\, a)}$, Equation 5.4 is then immediate from injectivity of $\flat$. □

*Remark* 5.51. Unlike for the envelope duploid, Lemma 5.50 is insufficient to make $\iota_{\mathcal{E}}^{\ominus}$ and $\iota_{\mathcal{E}}^{\oplus}$ equivalences on objects. This is because (say) a semantically negative object $\iota_{\mathcal{E}}^{\oplus}\, p : \mathcal{E}[L \dashv_{(\eta,\epsilon)} R]^{\ominus}$ for some $p : \mathcal{P}$ need only have $RL\eta_p = \eta_{RLp}$ (Lemma 5.47), which does not necessarily imply that $\eta_p$ has an inverse.

**Lemma 5.52** (`split_essentially_surjective_negative_category_to_envelope_duploid`). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The inclusions $\iota_{\mathcal{E}}^{\ominus} : \mathcal{N} \to \mathcal{E}[L \dashv R]_l^{\ominus}$ and $\iota_{\mathcal{E}}^{\oplus} : \mathcal{P} \to \mathcal{E}[L \dashv R]_t^{\oplus}$ are essentially surjective.*

*Proof.* The pre-image of $b : \mathcal{E}[L \dashv R]_l^{\ominus}$ in $\iota_{\mathcal{E}}^{\ominus}$ is $\pi_{\mathcal{E}}^{\ominus}\, b : \mathcal{N}$. The isomorphism in $\mathcal{E}[L \dashv R]_l^{\ominus}$ is $\iota_{\mathcal{E}}^{\ominus}(\pi_{\mathcal{E}}^{\ominus}\, b) \equiv \Uparrow b \overset{\mathtt{force}_b}{\cong_{lt}} b$. The case is dual for $\iota_{\mathcal{E}}^{\oplus}$. □

**Theorem 5.53** (`fully_faithful_negative_category_to_envelope_duploid_iff`, ...). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The following are equivalent:*

1. *The functor $\iota_{\mathcal{O}}^{\ominus} : \mathcal{N} \to \mathcal{O}[L \dashv R]_l^{\ominus\omega}$ is fully-faithful.*
2. *———————————— is an isomorphism of categories.*
3. *The functor $\iota_{\mathcal{E}}^{\ominus} : \mathcal{N} \to \mathcal{E}[L \dashv R]_l^{\ominus}$ is fully-faithful.*
4. *———————————— is a strong equivalence of categories.*

[2] NOTE because my rendition of the Fundamental Theorem of Identity Types only gives me biased reflexive graph induction lol.

5. *For all $n : N$, the fork $LRLRn \overset{LR\epsilon_n}{\underset{\epsilon_{LRn}}{\rightrightarrows}} LRn \xrightarrow{\epsilon_n} n$ is a coequalizer:*

$$\begin{tikzcd} LRLRn \arrow[r, shift left, "LR\epsilon_n"] \arrow[r, shift right, "\epsilon_{LRn}"'] & LRn \arrow[r, "\epsilon_n"] \arrow[dr, "f"'] & n \arrow[d, dashed, "\exists! f^*"] \\ & & m \end{tikzcd}$$

*That is, for all objects $n, m : \mathcal{N}$ and morphisms $f : LRn \to m$ satisfying $LR\epsilon_n ⨟ f = \epsilon_{LRn} ⨟ f$, there is a unique morphism $f^* : n \to m$ such that $\epsilon_n ⨟ f^* = f$.*

*Dually, the following are equivalent:*

1. *The functor $\iota^{\oplus}_{\mathcal{O}} : \mathcal{N} \to \mathcal{O}[L \dashv R]^{\oplus\omega}_t$ is fully-faithful.*
2. ______________________ *is an isomorphism of categories.*
3. *The functor $\iota^{\oplus}_{\mathcal{E}} : \mathcal{N} \to \mathcal{E}[L \dashv R]^{\oplus}_t$ is fully-faithful.*
4. ______________________ *is a strong equivalence of categories.*
5. *For all $p : P$, the fork $q \xrightarrow[\eta_p]{} RLq \overset{RL\eta_q}{\underset{\eta_{RLq}}{\rightrightarrows}} RLRLq$ is an equalizer:*

$$\begin{tikzcd} p \arrow[d, dashed, "\exists! g^*"'] \arrow[dr, "g"] & & \\ q \arrow[r, "\eta_p"'] & RLq \arrow[r, shift left, "RL\eta_q"] \arrow[r, shift right, "\eta_{RLq}"'] & RLRLq \end{tikzcd}$$

*That is, for all objects $p, q : \mathcal{P}$ and morphisms $g : p \to RLq$ satisfying $g ⨟ RL\eta_q = g ⨟ \eta_{RLq}$, there is a unique morphism $g^* : p \to q$ such that $g^* ⨟ \eta_q = g$.*

*Proof.* The implications (2/B) ⇒ (1/A) and (4/D) ⇒ (3/C) are immediate. Their converses (1/A) ⇒ (2/B) and (3/C) ⇒ (4/D) follow from Lemma 5.49 and Lemma 5.52. The equivalences (1/A) ⇔ (3/C) ⇔ (5/E) follow from the characterizations of thunkable and of linear morphisms in terms of the unit $\eta$ and counit $\epsilon$ of the adjunction (Lemma 5.21 and Lemma 5.46). □

**Definition 5.54** (`is_negative_equalizing`, ..., `is_fully_equalizing` [23]). We say that an adjunction satisfying conditions 1-5 of Theorem 5.53 is *negative equalizing*; and that an adjunction satisfying conditions A-C is *positive equalizing*. We say that an adjunction which satisfies conditions 1-5 **and** conditions A-E is *fully equalizing*.

*Remark* 5.55. By other authors, [3, 17] a functor $R\colon \mathcal{P} \to \mathcal{N}$ is known as being of *descent type* when it is the right adjoint in a negative equalizing adjunction. Likewise, a functor $L\colon \mathcal{N} \to \mathcal{P}$ is of *codescent type* when it is the left adjoint in a positive equalizing adjunction.

**Theorem 5.56** (is_univalent_split_oblique_duploid, is_univalent_split_oblique_-duploid_inv)**.** *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be a fully equalizing adjunction. The oblique duploid $\mathcal{O}[L \dashv R]$ is univalent as a split duploid if and only if $\mathcal{P}$ and $\mathcal{N}$ are univalent.*

*Proof.* Recall that univalence as a split duploid means that $\mathcal{O}[L \dashv R]_l^{\ominus\omega}$ and $\mathcal{O}[L \dashv R]_l^{\ominus\omega}$ are univalent. Univalence of $\mathcal{N}$ and $\mathcal{P}$ may be transported across the isomorphisms $\iota^{\ominus}_{\mathcal{O}} : \mathcal{N} \cong \mathcal{O}[L \dashv R]_l^{\ominus\omega}$ and $\iota^{\oplus}_{\mathcal{O}} : \mathcal{P} \cong \mathcal{O}[L \dashv R]_t^{\oplus\omega}$. □

**Lemma 5.57** (is_negative_fixed_point_iff_pre_fixed_point, …)**.** *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. If $L \dashv R$ is negative equalizing, then every object $n : \mathcal{N}$ satisfying*

$$LR\epsilon_n = \epsilon_{LRn}$$

*has that $\epsilon_n$ is an isomorphism. Dually, if $L \dashv R$ is positive equalizing, then every object $p : \mathcal{N}$ satisfying*

$$RL\eta_p = \eta_{RLp}$$

*has that $\eta_p$ is an isomorphism.*

*Proof.* (sketch) This follows from the universal coequalizer and equalizer conditions of Theorem 5.53, instantiated with $f := \mathsf{id}_{LRn}$ and $g := \mathsf{id}_{RLq}$ respectively. □

**Lemma 5.58** (envelope_chosen_negative_weq_is_negative)**.** *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be a fully equalizing adjunction. Every negative object in $\mathcal{E}[L \dashv R]$ has a negative polarization choice, and every positive object in $\mathcal{E}[L \dashv R]$ has a positive polarization choice.*

*Proof.* We show the former. Let $a : \mathcal{E}[L \dashv R]^{\ominus}$ be a negative object. Being an envelope object, $a$ has some polarization choice; if this is negative then we are done, otherwise assume that $a$ has a positive polarization choice. This means that $\mathsf{cross}\, a^{\flat}$ is an isomorphism, and (by Lemma 5.39) that $a$ is also positive. We therefore have that $\Downarrow a$, being linearly-and-thunkably isomorphic to $a$, is positive, and so $\eta_{\pi^{\oplus} a}$ is an isomorphism. Finally, it follows that

$$\mathsf{cross}\, a^{\sharp} = \varphi_{(\pi^{\oplus} a),(\pi^{\ominus} a)}(\mathsf{cross}\, a^{\flat}) = \eta_{\pi^{\oplus} a} \fatsemi \mathsf{cross}\, a^{\flat},$$

being a composite of isomorphisms, is itself an isomorphism. □

**Lemma 5.59** (`is_catiso_negative_category_to_envelope_duploid`, …). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be a fully equalizing adjunction between* ***univalent*** *categories $\mathcal{P}$ and $\mathcal{N}$. The inclusions $\iota^{\ominus}_{\mathcal{E}} \colon \mathcal{N} \to \mathcal{E}[L \dashv R]^{\ominus}_{l}$ and $\iota^{\oplus}_{\mathcal{E}} \colon \mathcal{P} \to \mathcal{E}[L \dashv R]^{\oplus}_{t}$ are isomorphisms of categories*

*Proof.* Combine Lemma 5.58 and Lemma 5.50. □

**Theorem 5.60** (`is_univalent_envelope_duploid`). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be a fully equalizing adjunction between* ***univalent*** *categories $\mathcal{P}$ and $\mathcal{N}$. The envelope duploid $\mathcal{E}[L \dashv R]$ is univalent.*

*Proof.* Analogous to Theorem 5.56, using Theorem 4.17 and Lemma 5.59. □

# 6. TODO Conclusions and Future Work

# Bibliography


[1] B. Ahrens, K. Kapulkin, and M. Shulman, "Univalent categories and the rezk completion," *Mathematical Structures in Computer Science*, vol. 25, no. 5, pp. 1010–1039, 2015. DOI: 10.1017/S0960129514000486

[2] B. Ahrens, P. R. North, M. Shulman, and D. Tsementzis, *The Univalence Principle* (Memoirs of the American Mathematical Society). 2025, vol. 305. DOI: 10.1090/memo/1541 arXiv: 2102.06275.

[3] M. Barr and C. Wells, *Toposes, Triples and Theories.* Springer New York, 1985, ISBN: 978-1-4899-0021-0.

[4] A. Church, "A formulation of the simple theory of types," *Journal of Symbolic Logic*, vol. 5, no. 2, pp. 56–68, 1940. DOI: 10.2307/2266170

[5] P. Clairambault and G. Munch-Maccagnoni, "Duploid situations in concurrent games," in *12th Workshop on Games for Logic and Programming Languages (GaLoP XII)*, Juha Kontinen, Marina Lenisa, Uppsala, Sweden, Apr. 2017. [Online]. Available: https://inria.hal.science/hal-01991555

[6] T. Coquand and N. A. Danielsson, "Isomorphism is equality," *Indagationes Mathematicae*, vol. 24, no. 4, pp. 1105–1120, 2013, In memory of N.G. (Dick) de Bruijn (1918–2012), ISSN: 0019-3577. DOI: 10.1016/j.indag.2013.09.002 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019357713000694

[7] M. H. Escardó, *A self-contained, brief and complete formulation of voevodsky's univalence axiom*, 2018. arXiv: 1803.02294 [math.LO]. [Online]. Available: https://arxiv.org/abs/1803.02294

[8] C. Führmann, "Direct models of the computational lambda-calculus," *Electronic Notes in Theoretical Computer Science*, vol. 20, pp. 245–292, 1999, MFPS XV, Mathematical Foundations of Progamming Semantics, Fifteenth Conference, ISSN: 1571-0661. DOI: 10.1016/S1571-0661(04)80078-1 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1571066104800781

[9] D. R. Grayson, “An introduction to univalent foundations for mathematicians,” *Bull. Amer. Math. Soc.*, vol. 55, pp. 427–450, 2018, ISSN: 1088-9485. DOI: 10.1090/bull/1616

[10] D. R. Grayson et al., *Unimath/unimath: V20250923*, version v20250923, Sep. 2025. DOI: 10.5281/zenodo.17186647

[11] M. Hofmann and T. Streicher, “The groupoid model refutes uniqueness of identity proofs,” in *Proceedings Ninth Annual IEEE Symposium on Logic in Computer Science*, 1994, pp. 208–212. DOI: 10.1109/LICS.1994.316071

[12] C. Kapulkin, P. L. Lumsdaine, and V. Voevodsky, *Univalence in simplicial sets*, 2018. arXiv: 1203.2553 [math.AT]. [Online]. Available: https://arxiv.org/abs/1203.2553

[13] K. Kapulkin and P. L. Lumsdaine, “The simplicial model of univalent foundations (after voevodsky),” *Journal of the European Mathematical Society*, vol. 23, no. 6, pp. 2071–2126, Mar. 2021, ISSN: 1435-9863. DOI: 10.4171/jems/1050

[14] J. Lambek and P. J. Scott, *Introduction to higher order categorical logic*. USA: Cambridge University Press, 1986, ISBN: 0521246652.

[15] P. B. Levy, “Adjunction models for call-by-push-value with stacks,” *Electronic Notes in Theoretical Computer Science*, vol. 69, pp. 248–271, 2003, CTCS’02, Category Theory and Computer Science, ISSN: 1571-0661. DOI: 10.1016/S1571-0661(04)80568-1

[16] P. B. Levy, “Call-by-push-value: A subsuming paradigm,” in *Typed Lambda Calculi and Applications*, J.-Y. Girard, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 228–243, ISBN: 978-3-540-48959-7.

[17] P. B. Levy, “Contextual isomorphisms,” *SIGPLAN Not.*, vol. 52, no. 1, pp. 400–414, Jan. 2017, ISSN: 0362-1340. DOI: 10.1145/3093333.3009898

[18] É. Mangel, P.-A. Melliès, and G. Munch-Maccagnoni, “Classical notions of computation and the hasegawa-thielecke theorem,” *Proc. ACM Program. Lang.*, vol. 10, no. POPL, Jan. 2026. DOI: 10.1145/3776715

[19] P. Martin-Löf, “Constructive mathematics and computer programming,” in *Logic, methodology and philosophy of science, VI (Hannover, 1979)*, ser. Stud. Logic Found. Math. Vol. 104, North-Holland, Amsterdam, 1982, pp. 153–175. DOI: 10.1016/S0049-237X(09)70189-2

[20] E. Moggi, "Computational lambda-calculus and monads," in *[1989] Proceedings. Fourth Annual Symposium on Logic in Computer Science*, 1989, pp. 14–23. DOI: 10.1109/LICS.1989.39155

[21] E. Moggi, "Notions of computation and monads," *Information and Computation*, vol. 93, no. 1, pp. 55–92, 1991, Selections from 1989 IEEE Symposium on Logic in Computer Science, ISSN: 0890-5401. DOI: 10.1016/0890-5401(91)90052-4

[22] L. de Moura and S. Ullrich, "The lean 4 theorem prover and programming language," in *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings*, Berlin, Heidelberg: Springer-Verlag, 2021, pp. 625–635, ISBN: 978-3-030-79875-8. DOI: 10.1007/978-3-030-79876-5_37 [Online]. Available: https://doi.org/10.1007/978-3-030-79876-5_37

[23] G. Munch-Maccagnoni, "Models of a non-associative composition," in *Foundations of Software Science and Computation Structures*, A. Muscholl, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 396–410, ISBN: 978-3-642-54830-7. DOI: 10.1007/978-3-642-54830-7_26

[24] G. Munch-Maccagnoni, "Syntax and Models of a non-Associative Composition of Programs and Proofs," Theses, Université Paris-Diderot - Paris VII, Dec. 2013. [Online]. Available: https://theses.hal.science/tel-00918642

[25] E. Rijke, *Introduction to Homotopy Type Theory* (Cambridge Studies in Advanced Mathematics). Cambridge University Press, 2025.

[26] T. Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. [Online]. Available: https://homotopytypetheory.org/book

# Index

# A. Counterexample Unital Magmoids

In this section I include some small examples of unital magmoids, to show that certain properties obtained in categories and preduploids do not generalize to unital magmoids.

**Example A.1.** There is a unital magmoid **Lini** (`lini`) that has a linear-and-thunkable isomorphism $p : b \cong_{lt} d$ which is not intermediate. **Lini** is generated by four objects: $a, b, c, d : \mathbf{Lini}$ and the morphisms

$$a \xrightarrow{f} b \xrightarrow{g} c$$
$$b \xrightarrow{p}_{lt} d \xrightarrow{q}_{lt} b$$

such that $p \fatsemi q = \mathsf{id}_b$ and $q \fatsemi p = \mathsf{id}_a$. In other words, $p$ and $q$ form a linear-and-thunkable isomorphism $p : b \cong_{lt} d$ with inverse $p^{-1} = q$. This is depicted in the diagram

$$\begin{tikzcd} a \arrow[r, "f"] & b \arrow[r, "g"] \arrow[d, bend right, "p"'] & c \\ & d \arrow[u, bend right, "q"'] & \end{tikzcd} .$$

The unital magmoid **Lini** can be described explicitly by setting

$$\mathbf{Lini}\llbracket a, c \rrbracket \overset{\text{def}}{\equiv} \{[f\,;g], [[f\,;p]\,;[q\,;g]]\} \quad (\text{or } \texttt{bool}).$$

All other hom-sets are either `unit` or `empty`, as needed to satisfy the existence of $f$, $g$, $p$, $q$, identities, and composites. The element $[f\,;g] : \mathbf{Lini}\llbracket a, c \rrbracket$ is just notation for, say `false` : `bool`. Composition is uniquely determined by unitality and the equations

$$f \fatsemi g \overset{\text{def}}{\equiv} [f\,;g] \quad \text{and} \quad (f \fatsemi p) \fatsemi (q \fatsemi g) \overset{\text{def}}{\equiv} [[f\,;p]\,;[q\,;g]]\,.$$

It may be computed that $p$ is a linear-and-thunkable isomorphism $b \cong_{lt} d$. Moreover, $p$ does not satisfy $(f \fatsemi p) \fatsemi (q \fatsemi g) = f \fatsemi g$, and (therefore) is not intermediate.

**Example A.2.** There is a finite unital magmoid **Inu** (`inu`) with a morphism $f\colon a \to b$

that has two inverses. **Inu** is generated by the three morphisms

$$
\begin{tikzcd}
a \arrow[r, "f"] & b \arrow[l, bend right, "p"'] \arrow[l, bend left, "q"]
\end{tikzcd}
$$

and the equations

$$
\begin{aligned}
f \mathbin{⨾} p = \mathsf{id}_a = f \mathbin{⨾} q \\
p \mathbin{⨾} f = \mathsf{id}_b = q \mathbin{⨾} f\,.
\end{aligned}
$$

This may be constructed explicitly by setting

$$
\mathbf{Inu}[\![b, a]\!] \stackrel{\text{def}}{\equiv} \{p, q\} \quad (\text{or } \texttt{bool}),
$$

and all other hom-sets to `unit`.

Additionally, it can be computed that $f$ is linear and thunkable but not intermediate, and that $p$ and $q$ are both intermediate but not linear or thunkable.