UNIVERSITY OF CAMBRIDGE
Department of Computer Science and Technology

Univalent Models of a Non-Associative Composition

Beatrice Szilvasy

St Catharine's College

May 22, 2026

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

Total page count: 67

Main chapters (excluding front-matter, references and appendix): 49 pages (pp 1–49)

Main chapters word count: 11377 (text: 10002, headers: 63, captions/footnotes: 221, inlinemath: 1008, displaymath: 83)

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 back-matter 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

Mixing call-by-value and call-by-name evaluation orders in a single programming language gives rise to a composition which is non-associative. To model their semantics, we may instead use Munch-Maccagnoni's *duploids*, category-like structures that relax the associativity of composition. We study and formalize duploids in homotopy type theory/univalent foundations. Our work includes a definition of what it means for a duploid to be *univalent*: that isomorphisms of a suitable kind render the objects indistinguishable to the type theory. We provide a novel construction for the duploid arising from an adjunction, that is univalent in a sense which the existing construction is not. This work is formalized in the UniMath library for the Rocq proof assistant.

Acknowledgements

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

Contents

1. Introduction

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

$$\texttt{let } \left(\texttt{let } e_1 \texttt{ be } x \texttt{ in } e_2\right) \texttt{ be } y \texttt{ in } e_3 \quad \text{and} \tag{1.1}$$

$$\texttt{let } e_1 \texttt{ be } x \texttt{ in } \left(\texttt{let } e_2 \texttt{ be } y \texttt{ in } e_3\right) \tag{1.2}$$

$$(\text{where } e_3 \text{ does not reference } x)$$

may compute differently. 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:

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

Here, evaluating $\texttt{print}(v)$ outputs 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 $\texttt{let } e_1 \texttt{ be } x \texttt{ in } e_2$ by first substituting e_1 for x in e_2. Then evaluate the resulting expression $e_2[x := e_1]$. Using call-by-name, Program 1.3 outputs `hellohello` and returns 2.

- Using *call-by-value* (CBV), evaluate $\texttt{let } e_1 \texttt{ be } x \texttt{ in } e_2$ 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-name or call-by-value, composition will remain associative. Combining call-by-name and call-by-value in a single language exposes non-associative composition.

Let us write

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

$$\texttt{let}\ e_1 \overset{\mathrm{V}}{\texttt{be}}\ x\ \texttt{in}\ e_2 \qquad \text{for call-by-value.}$$

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

$$\begin{gathered}\texttt{let}\ \Big(\texttt{let}\ (\texttt{print("hello")};\ 1)\overset{\mathrm{V}}{\texttt{be}}\ x\ \texttt{in} \\ (\texttt{print("world")};\ x)\Big) \\ \overset{\mathrm{N}}{\texttt{be}}\ y\ \texttt{in}\ y+y\end{gathered} \tag{1.4}$$

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

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

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 composition. Consider $\texttt{let}\ (\texttt{let}\ e_1\ \texttt{be}\ x\ \texttt{in}\ e_2)\ \texttt{be}\ y\ \texttt{in}\ e_3$ (Program 1.1) and $\texttt{let}\ e_1\ \texttt{be}\ x\ \texttt{in}\ (\texttt{let}\ e_2\ \texttt{be}\ y\ \texttt{in}\ e_3)$ (Program 1.2) to be interpreted in some category $\mathcal{C}$, and assign to each expression $x : A \vdash e : B$ a morphism $[\![e]\!] : [\![A]\!] \to [\![B]\!]$ in $\mathcal{C}$. The question of whether Programs 1.1 and 1.2 compute differently then amounts, informally,[1] 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 in any category $\mathcal{C}$. If we wish to take seriously the idea that equations of the form in Equation 1.6

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

do not necessarily hold, then we must use a different, more general model. In this dissertation, it will be Munch-Maccagnoni's *duploids* [25] that generalize and take the role of categories for modelling non-associative computation.

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* [18] (CBPV). The denotational semantics of call-by-push-value can be modelled "indirectly" by adjunctions [17], generalizing the widespread use of monads for the semantics of call-by-value [22]. Duploids fit into the picture by providing "direct" semantics, as a single category-like structure, for mixed evaluation order, and have a correspondence with adjunctions [25].

1.1. Organization and Contributions

The purpose of this dissertation is to develop the theory of duploids formally in a variant of Martin Löf type theory [21] and from a *univalent point of view* [29]. We[2] formalize some existing results about duploids within the Rocq [28] library UniMath [11]. **The starting point** is Munch-Maccagnoni [25]'s PhD thesis on duploids [25, Chapter 2 of 26], and the existing formalization of univalent categories in UniMath [2, 5, 11]. **Our contribution** is a formalization and study of duploids in univalent foundations. 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 formalize two constructions of the duploid arising from an adjunction; the first is Munch-Maccagnoni's own, and the second is a novel construction. We show both to be "univalent," but in different senses. As an overview:

1. In the Background chapter, we start with the context and motivation of duploids in Section 2.1. We introduce some preliminary definitions for homotopy type theory/univalent foundations in Section 2.2, and the necessary background on univalent category theory in Section 2.3.

2. In the Unital Magmoids chapter, we introduce a non-associative version of categories called *unital magmoids*. We introduce Munch-Maccagnoni's notions of polarity, and define what it means for a unital magmoid to be univalent.

3. In the Duploids chapter, we introduce two versions of duploids, *single-sorted duploids* and *split duploids*, and state what it means for each to be univalent.

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

4. In the Duploids from Adjunctions chapter, we define two versions of *the duploid arising from an adjunction*: the original "oblique duploid" construction of Munch-Maccagnoni [25] which is naturally a split duploid, and a novel "envelope duploid" construction which is a single-sorted duploid. We finish in Section 5.3 by showing that these are univalent, in their respective senses, under reasonable conditions.

Convention 1.1. Technical terms are set in *italics* when introduced, 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 "**Definition 3.1** (`category`). A *category* … ." Often a theorem or definition is associated with more than one formalized definition. In this case one or multiple may appear next to the statement, with similar or dual names elided with "*etc.*" or "and dual" for brevity: for example "**Definition 3.19** (`is_thunkable`, `is_negative`, *etc.*)." In anonymized versions, these links will necessarily be broken.

Convention 1.2. 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 definitions, statements, or results.

1.2. Related Work

This dissertation is primarily based on Munch-Maccagnoni's "Models of a Non-Associative Composition" [25, Chapter 2 of 26], which introduces duploids as a mathematical structure. Much of this work is based on formalizing the definitions and proofs of Munch-Maccagnoni [25] in the proof assistant Rocq. The parts that we do not include are the internal language of a duploid, and the *structure theorem* relating the category of duploids to the category of adjunctions. We note that in univalent foundations there is no **1**-category of small duploids, for the same reason there is no 1-category of small categories (see Non-example 3.14). The structure theorem for a 2-category of duploids is yet to appear in any literature.

The rest of Munch-Maccagnoni's PhD thesis [26] explores non-associative composition and polarization more broadly, relating it to programs, with first-class continuations/continuation-passing-style translations, and to proofs, with the involutive negation in classical logic. Other work on duploids includes that of Mangel et al. [20] about linear classical logic, which also includes some general theory about duploids. We note that our "correct notion of isomorphism" in a unital magmoid (Definition 3.28) differs from that of Mangel et al. [20], but they do coincide for preduploids (see Remark 3.30 and Corollary 4.14).

For the "univalent" part, this dissertation is based on the study of category theory in univalent foundations initiated by Ahrens et al. in "Univalent categories and the Rezk completion" [2]. Our work is also based on *op. cit.* in the literal sense that we directly use its Rocq formalization, which is today part of UniMath [11]. There are other works that formulate and formalize "univalently" some areas of category theory. Examples include bicategories [1], monoidal categories [31] and monads [30].

Another work is *The Univalence Principle* by Ahrens et al. [3], deriving algorithmically what it means for a large class of mathematical structures to be "univalent." Ahrens et al. mention duploids, but beware that they use a different definition to the one in this dissertation (see Remark 4.4). Still, our definitions of "univalent duploid" (Definitions 4.15 and 4.19) are the same as derived by their procedure.

Mixed evaluation order is also related to Levy's call-by-push-value [18]. Especially pertinent are Levy's "Contextual isomorphisms." Levy defines *contextual isomorphism* of types $\theta : A \cong B$ **syntactically** with respect to an equivalence relation $\sim$ on typed terms. Given a typing judgement with a type hole $\Gamma[-] \vdash C[-]$, such a contextual isomorphism bijectively (and naturally) maps $\sim$-classes of typed terms $\Gamma[A] \vdash t : C[A]$ to $\sim$-classes of typed terms $\Gamma[B] \vdash \theta(t) : C[B]$. Levy shows for call-by-push-value that when $\sim$ includes computation rules, contextual isomorphisms correspond to thunkable (Führmann [9]) isomorphisms for value (positive) types, and linear (Munch-Maccagnoni [25]) isomorphisms for computation (negative) types. This is similar to our definition of univalence for duploids, and our proofs of univalence in Section 5.3.

2. Background

2.1. Semantics and Evaluation Order

It is a well-established tradition in computer science to consider programs as terms in variants of the lambda calculus. In the study of their denotational semantics, we assign to each term a mathematical object that captures its computational behaviour. For the *pure* simply typed lambda calculus, evaluation order does not matter, and it is the *cartesian closed categories* that model their semantics [16]. However, many languages include *(computational) effects* such as input/output, mutable state, or non-termination. Modelling their semantics requires choices of evaluation order. Models include monads for call-by-value [22, 23], comonads for call-by-name [17, 25], adjunctions for call-by-push-value [17], and duploids for a mixed ("polarized") evaluation order [25, Chapter 1 of 26]. In this section we cover this context.

A simply typed lambda calculus term $\Gamma \vdash e : A$ may be interpreted in a cartesian closed category as a morphism $[\![e]\!] : [\![\Gamma]\!] \to [\![A]\!]$ by recursion on the typing judgement [16]. 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).

Moggi pointed out that the pure lambda calculus and its $\beta\eta$-equivalence are overly simplistic for program equivalence in the presence of effects, and suggested the use of monads to model effectful call-by-value lambda calculi instead [22, 23]. Given a suitable monad $T : \mathcal{C} \to \mathcal{C}$ 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 $\mathcal{C}$. Dually, one may use comonads to model call-by-name lambda calculi [17, 25]. Given a suitable comonad $U : \mathcal{D} \to \mathcal{D}$ on a category $\mathcal{D}$, we can interpret effectful call-by-name lambda terms $\Gamma \vdash e : A$ as morphisms $[\![e]\!] : U\,[\![\Gamma]\!] \to [\![A]\!]$.

Moggi's interpretation into monads 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 [9], monads constitute *indirect semantics* for the computational lambda calculus, in contrast to *direct semantics* such as cartesian closed categories for the pure lambda calculus. In direct semantics, the syntax is more closely related to the constructs readily available, whereas indirect semantics require an extra transformation step. Führmann [9] provides direct semantics for the computational lambda calculus which he calls *abstract Kleisli categories*, but which we shall call *thunk-force categories* [25]. These interpret terms $x : A \vdash e : B$ as morphisms $[\![e]\!] : [\![A]\!] \to [\![B]\!]$ in a single category of computations. All monads give rise to a thunk-force category by its *Kleisli category*, and all thunk-force categories are equivalent to one that arises this way. The call-by-name dual of a thunk-force category is a *runnable monad* [25].

The models mentioned so far presuppose a fixed evaluation order. However, as argued by Levy [18], the combination of call-by-name and call-by-value into a single subsuming paradigm, *call-by-push-value*, obviates the need to reason about both separately. Call-by-push-value has indirect semantics in adjunctions $\varphi_{p,n} : \mathcal{N}[\![Lp, n]\!] \simeq \mathcal{P}[\![p, Rn]\!]$ [17]. The syntactically-distinct value types V and computation types $\underline{C}$, are interpreted in $\mathcal{P}$ and $\mathcal{N}$ respectively. Then, call-by-push-value terms $x : V \vdash e : \underline{C}$, which produce computations from values, are interpreted as "oblique" morphisms $[\![e]\!] : [\![V]\!] \to R\,[\![\underline{C}]\!]$ in $\mathcal{P}$, or equivalently (by φ), $[\![e]\!] : L\,[\![V]\!] \to [\![\underline{C}]\!]$ in $\mathcal{N}$. These semantics are "subsuming": the translation of call-by-name and call-by-value into call-by-push-value, when interpreted in the adjunction semantics, gives rise to the corresponding comonad and monad semantics of the source.

Munch-Maccagnoni introduces direct semantics for mixed evaluation order in the form of *duploids* [25]. The goal is still to interpret terms $x : A \vdash e : B$ as morphisms $[\![e]\!] : [\![A]\!] \to [\![B]\!]$ in a single category-like structure. However, due to the non-associativity of composition with mixed call-by-name and call-by-value, duploids cannot require associativity like categories. Instead, duploids are instances of *unital magmoids*: categories minus the associativity law.

Duploids are closely related to adjunctions, in that each adjunction $L \dashv R$ gives rise to a duploid $\mathcal{D}$, and all duploids are equivalent to one that arises this way [25]. Moreover, this duploid $\mathcal{D}$ contains as subcategories—that is, as sub- unital magmoids of $\mathcal{D}$ which are associative—the Kleisli and co-Kleisli categories associated to the adjunction. In this way, duploids subsume both monads for call-by-value and comonads for call-by-name.

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

2.2. Preliminary Type Theory

Chapters 3–5 consist of an informal English 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. This section includes some notation and terminology we shall 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 [10]. For textbook accounts, I suggest Rijke's *Introduction to Homotopy Type Theory* [27], or the well-known *HoTT Book* [29]. Though insufficient for understanding this dissertation, Escardó has a concise self-contained introduction to the univalence axiom itself [8]. Knowledge of basic dependent type theory is assumed from here on out.

Convention 2.1. As in the UniMath [11] formalization, we work in Martin-Löf type theory [21] with intensional identity types and no W-types. Unlike the UniMath formalization, we do not assume in this text the existence of universes (Definition 2.9) 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 $(\lambda x : B.\, e)\ b \equiv a[x := b] : E[b]$ for $b : B$ and $x : B \vdash e : E[x]$; 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 equality of 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 an 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 all types 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*. Extending 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 $f\ x\ x\ \mathsf{refl}_x : E[x, x, \mathsf{refl}_x]$. More typically, given $p : e_1 =_A e_2$ we will informally use substitution to replace occurrences of e_1 with e_2 or vice-versa, particularly when the specific p is not relevant. 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

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

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 (higher) 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-types* 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 *centre* element $\mathsf{centre}_A : A$ and for all elements $a : A$ there is a chosen identification $a =_A \mathsf{centre}_A$.

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

Examples 2.7. The `empty` type is a proposition, and the `unit` type 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 $\texttt{iscontr}(A)$, $\texttt{isaprop}(A)$ and $\texttt{isaset}(A)$ 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 $\texttt{iscontr}(\sum_{x:B} E[x])$.

[1]This is orthogonal to matters of size. See Definition 2.10 for that.

Sometimes the data of a type A—which need not be a proposition—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

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

and for all propositions P an eliminator

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

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

Universes and Univalence

In our description so far, types have been treated separately from terms, since there has been no type of types. We remedy this now: a "large" type of "small" types is called a *universe*.

Definition 2.9. A *universe* $(\mathcal{U}, \mathsf{El})$ is a type $\mathcal{U}$ of *codes* along with, for each element $A : \mathcal{U}$, a decoding $\mathsf{El}(A)$ type. We shall typically abuse notation and use simply $\mathcal{U}$ to refer to the universe $(\mathcal{U}, \mathsf{El})$.

Definition 2.10. 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 $\mathsf{El}(\mathsf{code}(A)) \equiv A$ type. We shall typically omit El and code, so that $A : \mathcal{U}$ implies A type, and vice-versa when A is $\mathcal{U}$-small.

Convention 2.11. We may simply say that a type is small when the universe it is small with respect to are obvious or have not given a name. This rule will also apply to other things that may be "small."

Example 2.12. 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 $\mathsf{El}'\langle A, h_A\rangle \overset{\text{def}}{\equiv} \mathsf{El}(A)$. The universe $\mathtt{hSet}_{\mathcal{U}}$ is closed under all type constructors mentioned so far.

Crucially, a universe $\mathcal{U}$ allows forming the identification type $A =_{\mathcal{U}} B$ between two $\mathcal{U}$-small types A and B. What types should be identified, though? The types $\mathtt{unit} \amalg \mathtt{unit}$ and $\mathtt{bool}$ are essentially the same, but refl is insufficient to identify them. The *univalence axiom* provides a way to characterize identifications $A =_{\mathcal{U}} B$ as *equivalences* that justify A and B to be "essentially the same."

Definition 2.13. 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 $f^{-1}\colon B \to A$ 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]

Example 2.14. The identity function $\mathsf{id}_A\colon A \to A$, defined $\mathsf{id}_A\ x \stackrel{\text{def}}{\equiv} x$, is an equivalence.

Definition 2.15. In every universe $\mathcal{U}$, there is a canonical map

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

A *univalent universe* $\mathcal{U}$ is one where $\mathtt{id\_to\_weq}\ A\ B$ is an equivalence for all $\mathcal{U}$-small types $A, B : \mathcal{U}$. For a given universe $\mathcal{U}$, the *univalence axiom* states that $\mathcal{U}$ is univalent.

Example 2.16. If a universe $\mathcal{U}$ is univalent, then so too is $\mathtt{hSet}_{\mathcal{U}}$.

The type of equivalences between types is not in general a proposition. Both $\mathsf{id}_{\mathtt{bool}}$ and boolean negation are equivalences $\mathtt{bool} \simeq \mathtt{bool}$, for example. Thus, the univalence axiom provides a way of constructing identifications $x =_A x$ which are provably **not** identifiable with refl_x. This property is perfectly consistent with J [12], but not with equality reflection. Indeed there are models of type theory including a univalence axiom that are as consistent as ZFC with some strongly inaccessible cardinals [13, 14, 15].

[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.

Remark 2.17. The univalence axiom for a universe $\mathcal{U}$ provides that all type families $E[A]$ indexed by $A : \mathcal{U}$ are congruent with equivalences: that each $A \simeq B$ lifts to an equivalence $E[A] \simeq E[B]$. However, this principle may also be proven directly for many type families, such as $\mathtt{iscontr}(A)$ and $\mathtt{isaprop}(A)$. We shall therefore not actually need the univalence axiom ourselves for the main results of this paper.

Classical Mathematics and Logic

By contrast to "univalent foundations," I (ab)use the term *set-based foundations* to include ZF(C) and type theories where all types are sets, in the sense of Definition 2.6.[4] 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 $\mathtt{hSet}_{\mathcal{U}}$ instead of $\mathcal{U}$ for a fixed universe $\mathcal{U}$.

It is an occasional misconception 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 consistent 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.

Further Conventions

Convention 2.18. 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 of Table 2.1 suggests homotopical interpretations. These may be realized more formally by models of univalent foundations such as Voevodsky's in simplicial sets [14, 15].

Convention 2.19. 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 $(\alpha, \beta \colon F \Rightarrow G)$.

[4] This includes the type theory of the *Lean 4 Theorem Prover* [24].

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/*etc.* 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	$\Vert A \Vert$ (propositional truncation)	A is inhabited
there exists $x : B$ such that $P[x]$	$\left\Vert \sum_{x:B} P[x] \right\Vert$ 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 $\Vert A \amalg B \Vert$	
$f : A \to B$ 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

Notation 2.20. 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.

2.3. 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* [25]. Before we can talk about univalent duploids, we ought understand categories in the context of univalent foundations.

The authoritative paper on categories in univalent foundations is by Ahrens et al. [2], whose formalization in Rocq is today part of UniMath. An expanded version of that paper is included as Chapter 9 of the *HoTT Book* [29]. 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,[5] and there may be isomorphisms $a \cong a$ that are not the identity.[6] In univalent foundations, the type of identifications $a =_{\mathsf{ob}\,\mathcal{C}} b$ 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 $F\colon \mathcal{C} \to \mathcal{D}$ 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 $F\colon \mathcal{C} \to \mathcal{D}$ 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.

[5] Say, the sets $\{0, 1\}$ and $\{1, 2\}$.
[6] Say, the set $\mathbb{B}$ under boolean negation.

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. [7] that the univalence axiom extends to a "large class of algebraic structures." This implies that categories of set-based algebraic structures such as monoids and rings are univalent too.

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 $\mathsf{y} \colon \mathcal{C} \to \mathbf{Fn}[\mathcal{C}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}]$ [2].

The univalence of a category is sensitive to how the category is constructed. A relevant example is the *Kleisli category* of a monad $T \colon \mathcal{C} \to \mathcal{C}$ 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 $\mathcal{C}$. Even when $\mathcal{C}$ is univalent, this construction of the Kleisli category is **not** generally univalent (see Example 2.21). 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 $\mathbf{Kleisli}(\mathcal{C}, T)$. [TODO: This is an abrupt stop for the chapter. I can tie this in nicely, probably.]

Example 2.21 (Levy [19]). 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 $\mathtt{unit} \cong \mathtt{bool}$ in $\mathbf{Kleisli}_{\cancel{\mathsf{univ}}}(\mathbf{Set}, T)$: this is simply an isomorphism $\mathtt{unit} \times \mathbb{N} \cong \mathtt{bool} \times \mathbb{N}$ in $\mathbf{Set}$. However, the type $\mathtt{unit}$ is certainly not identified with $\mathtt{bool}$!

3. Unital Magmoids

In this chapter we study *unital magmoids*, a generalization of categories that drops the associativity of composition. We shall build on unital magmoids to study duploids in subsequent chapters. In Section 3.1, we review some necessary concepts of category theory, and how they generalize to unital magmoids. In Section 3.2, we introduce Munch-Maccagnoni's characterization of the concept of *polarity* in logic as the failure of associativity [25, 26]. In Section 3.3, we shall define and justify what a "univalent unital magmoid" is, as a generalization of univalent categories.

3.1. Definitions

Definition 3.1 (`category` [2])**.** 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 $\mathsf{id}_a : a \to a$.
4. A *composition* operator $\fatsemi_{\mathcal{C}}$ or simply $\fatsemi$ assigning to each pair of morphisms $f : a \to b$ and $g : b \to c$ a composite $f \fatsemi g : a \to c$ 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 (`unital_magmoid`)**.** A *unital magmoid* or *non-associative category* [20] is a generalization of a category that need not be associative. 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 id_a, and composition $f \mathbin{⨾} g$—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 (Remark 3.7) and hom-functors (Example 3.13).

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 $\mathcal{U}$-small and its type of objects $\mathsf{ob}\,\mathcal{M}$ is $\mathcal{U}$-small. Following Convention 2.11, we may omit the universe when describing unital magmoids and categories as "small."

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, but the directions of morphisms inverted:

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

The identity 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 $\mathcal{M}$.

Example 3.5. For two categories or unital magmoids $\mathcal{M}_1, \mathcal{M}_2$, their *product* is the category or unital magmoid $\mathcal{M}_1 \times \mathcal{M}_2$ with objects and morphisms pairs of those in $\mathcal{M}_1$ and $\mathcal{M}_2$:

$$\mathsf{ob}(\mathcal{M}_1 \times \mathcal{M}_2) \overset{\text{def}}{\equiv} \mathsf{ob}\,\mathcal{M}_1 \times \mathsf{ob}\,\mathcal{M}_2$$
$$(\mathcal{M}_1 \times \mathcal{M}_2)[\![\langle a_1, a_2\rangle, \langle b_1, b_2\rangle]\!] \overset{\text{def}}{\equiv} \mathcal{M}_1[\![a_1, b_1]\!] \times \mathcal{M}_2[\![a_2, b_2]\!].$$

Identities and composition are defined pointwise.

Definition 3.6 (`is_z_isomorphism`, `z_iso`)**.** A morphism $f\colon a \to b$ in a unital magmoid or category *has an inverse* if there is a chosen morphism $f^{-1}\colon b \to a$ such that

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

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

Remark 3.7 (isaprop_is_z_isomorphism). Inverses are unique in a category. That is, if we have a morphism $f : a \to b$ in a category and two morphisms $p, q : b \to a$ satisfying Equation 3.1, then we have $p = q$. The proof depends on associativity:

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

This makes the type "f has an inverse" be a proposition in a category. In a unital magmoid this is no longer true in general (see Example A.2), but we still use "isomorphism" to mean a morphism with a specified inverse.

Definition 3.8 (is_univalent [2]). Let $\mathcal{C}$ be a category. Then there is a canonical map from identifications to isomorphisms

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

The category $\mathcal{C}$ is *univalent* if id_to_iso a b is an equivalence for all $a, b : \mathcal{C}$.

Definition 3.9 (edges_from, edges_to). In a category $\mathcal{C}$, for an object $a : \mathcal{C}$, the type

$$\{a\}_{\mathcal{C}}^{\Rightarrow} \overset{\text{def}}{\equiv} \sum_{b:\mathcal{C}} a \cong b \qquad \text{is called the } \textit{outward fan} \text{ of } a \text{, and}$$

$$\{a\}_{\mathcal{C}}^{\Leftarrow} \overset{\text{def}}{\equiv} \sum_{b:\mathcal{C}} b \cong a \qquad \text{is called the } \textit{inward fan} \text{ of } a.$$

Theorem 3.10 (is_rxgraph_univalent_from_isaprop_edges_from [27, Chapter 11]). *Let $\mathcal{C}$ be a category. The following are equivalent.*

1. *The category $\mathcal{C}$ is univalent.*
2. *All outward (dually, inward) fans of $\mathcal{C}$ are propositions.*
3. *All outward (dually, inward) fans $\{a\}_{\mathcal{C}}^{\Rightarrow}$ are contractible with centre $\langle a, \mathsf{id}_a \rangle : \{a\}_{\mathcal{C}}^{\Rightarrow}$.*

Proof. This is a variant of Rijke's *fundamental theorem of identity types* [27].[1] □

[1]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.

We do not generalize the definition of "univalent category" to unital magmoids directly. Instead, we shall (in Definition 3.28) use a stronger notion of isomorphism for the univalence condition of unital magmoids, which simplifies to Definition 3.8 when the unital magmoid is a category.

As is typical in category theory, we are not only concerned with unital magmoids, but also with maps between unital magmoids, and maps betwen those maps. We import the concepts of *functors* and *natural transformations* directly, but also provide some necessary generalizations.

Definition 3.11 (`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 $F_1 \colon \mathcal{N}[\![a, b]\!] \to \mathcal{N}[\![F_0\, a, F_0\, b]\!]$.
3. F preserves:
   a) Identities:
   $$F_1\left(a \xrightarrow{\mathsf{id}_a} a\right) = F_0\ a \xrightarrow{\mathsf{id}_{F_0 a}} F_0\, a \qquad \text{for all } a : \mathcal{N}.$$
   b) Composition:
   $$F_1\left(a \xrightarrow{f} b \xrightarrow{g} c\right) = F_0\, a \xrightarrow{F_1 f} F_0\, b \xrightarrow{F_1 g} F_0\, c \qquad \text{for all } a \xrightarrow{f} b \xrightarrow{g} c \text{ in } \mathcal{N}.$$

Both F_0 and F_1 may simply be written F.

Definition 3.12 (`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 3.13. In a locally small category $\mathcal{C}$, there is a categorical functor

$$\mathcal{C}[\![-,-]\!] \colon \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set} \qquad \text{given by}$$
$$\mathcal{C}[\![-,-]\!]_0 \langle a, b\rangle \overset{\text{def}}{\equiv} \mathcal{C}[\![a, b]\!]$$
$$\mathcal{C}[\![-,-]\!]_1 \left\langle a \xrightarrow{f} b, c \xrightarrow{h} d\right\rangle \left(b \xrightarrow{g} c\right) \overset{\text{def}}{\equiv} a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d.$$

The same data gives only a reflexive graph functor $\mathcal{M}[\![-,-]\!] \colon \mathcal{M}^{\mathsf{op}} \times \mathcal{M} \dashrightarrow \mathbf{Set}$ in general for a unital magmoid $\mathcal{M}$. This is called the *hom-functor* of $\mathcal{C}$ or $\mathcal{M}$.

Non-example 3.14. Functions between categories or unital magmoids have identities and composition. We shall write these $\mathsf{Id}_{\mathcal{M}} \colon \mathcal{M} \to \mathcal{M}$, and $GF \colon \mathcal{M}_1 \to \mathcal{M}_3$ for $F \colon \mathcal{M}_1 \to \mathcal{M}_2$ and $G \colon \mathcal{M}_2 \to \mathcal{M}_1$ (in the opposite order to $⨾$). Despite satisfying the unitality and associativity laws of a category, $\mathcal{U}$-small categories and functors between them (for some universe $\mathcal{U}$) do **not** themselves form a category $\mathbf{Cat}_{\mathcal{U}}$. This is because the type of functors does not in general form a set.[2] One may restrict to *strict categories* $\mathbf{Cat}_{\mathtt{hSet}_{\mathcal{U}}}$—categories $\mathcal{C}$ whose objects $\mathsf{ob}\,\mathcal{C}$ form a set—but this yields a too-strict notion of equivalence between categories: isomorphism rather than adjoint equivalence. One ought to consider $\mathbf{Cat}_{\mathcal{U}}$ as a 2-category instead [1].

Definition 3.15 (`nat_trans`)**.** A *natural transformation* α between categorical or reflexive graph functors $F, G \colon \mathcal{N} \dashrightarrow \mathcal{M}$, 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 $\alpha_a ⨾ Gf = Ff ⨾ \alpha_b$ holds. That is, the *naturality square*

$$\begin{tikzcd} Fa \arrow[r, "Ff"] \arrow[d, "\alpha_a"'] & Fb \arrow[d, "\alpha_b"] \\ Ga \arrow[r, "Gf"] & Gb \end{tikzcd} \quad \text{in } \mathcal{M} \text{ commutes.} \tag{3.3}$$

Definition 3.16 (`nat_trans_data`)**.** An *unnatural transformation* $\alpha \colon F \rightleftharpoons G$ between categorical or reflexive graph functors $F, G \colon \mathcal{N} \dashrightarrow \mathcal{M}$, by contrast, consists only of a family of morphisms $\alpha_a \colon Fa \to Ga$, and need not satisfy naturality.

Example 3.17 (`functor_category`, `is_univalent_functor_category` [2])**.** Let $\mathcal{M}$ be a unital magmoid and $\mathcal{D}$ a category. There is a category $\mathbf{Fn}[\mathcal{M}, \mathcal{D}]$ whose objects are functors $F, G \colon \mathcal{M} \to \mathcal{D}$ and whose morphisms are natural transformations $F \Rightarrow G$. If $\mathcal{D}$ is univalent, then the functor category $\mathbf{Fn}[\mathcal{M}, \mathcal{D}]$ is univalent.

Example 3.18 (`rxfunctor_unital_magmoid`, `natrxfunctor_category`)**.** Example 3.17 does not generalize to unital magmoids, because the composite of natural transformations is not natural in general. Let $\mathcal{N}$ and $\mathcal{M}$ be unital magmoids. There is a unital magmoid $\mathbf{RxFn}[\mathcal{N}, \mathcal{M}]$ consisting of reflexive graph functors $F, G \colon \mathcal{N} \dashrightarrow \mathcal{M}$ and unnatural transformations $F \rightleftharpoons G$ between them. If $\mathcal{M}$ is a category, $\mathbf{RxFn}[\mathcal{N}, \mathcal{M}]$ is a category and has a subcategory $\mathbf{RxFn}_{\mathsf{nat}}[\mathcal{N}, \mathcal{M}]$ of reflexive graph functors and natural transformations.

[2] 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 composites involving certain objects or morphisms. In this section, we define the concept of linearity, thunkability, and polarization, or rather their characterizations in unital magmoids due to Munch-Maccagnoni [25]. We also coin the term *intermediate* for a less-studied notion similar to linearity and thunkability. We find use for it to define univalence of unital magmoids in Section 3.3.

Definition 3.19 (`is_thunkable`, `is_negative`, *etc.* [25])**.** 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 parenthesizing their composite $f_1 \mathbin{⨾} f_2 \mathbin{⨾} \cdots \mathbin{⨾} f_n$ ($= \mathsf{id}_{a_0}$ 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 $f \mathbin{⨾} (g \mathbin{⨾} h) = (f \mathbin{⨾} g) \mathbin{⨾} h$.

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.

$$\begin{array}{ccccccc} & & & \text{intermediate} & & & \\ & \text{thunkable} & & | & & \text{linear} & \\ & | & & | & & | & \\ a & \xrightarrow{f} & b & \xrightarrow{g} & c & \xrightarrow{h} & d \\ & & | & & | & & \\ & & \text{negative} & & \text{positive} & & \end{array}$$

Figure 3.1. The triple (f, g, h) associates if any of the labelled properties holds.

Notation 3.20. When a path of morphisms as in Diagram 3.4 is known to associate, we may write its composite without parentheses as $f_1 \mathbin{⨟} f_2 \mathbin{⨟} \cdots \mathbin{⨟} f_n$. Following Notation 2.20, we may 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 for the composite.

Remark 3.21. Thunkability/linearity and negativity/positivity are dual, while intermediateness is self-dual. That is, being thunkable/linear or negative/positive in some unital magmoid $\mathcal{M}$ is the same as being linear/thunkable or positive/negative in $\mathcal{M}^{\mathsf{op}}$.

Example 3.22. In a unital magmoid, the identity morphisms are thunkable, linear, and intermediate by unitality. Composites of thunkable, linear, or intermediate morphisms are also thunkable, linear, or intermediate respectively.

Definition 3.23 (`linear_category`, `negative_category`, *etc.* [25])**.** 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* $\mathcal{M}_l$ of linear morphisms.
- The *thunkable subcategory* $\mathcal{M}_t$ of thunkable morphisms.
- The *intermediate subcategory* $\mathcal{M}_i$ of intermediate morphisms.
- The (*semantically*) *negative subcategory* $\mathcal{M}^{\ominus}$ of semantically negative objects.
- The (*semantically*) *positive subcategory* $\mathcal{M}^{\oplus}$ of semantically positive objects.
- Combinations of the above, notably:
  - The *linear-and-thunkable subcategory* $\mathcal{M}_{lt}$ of linear and thunkable morphisms.
  - The *linear-and-thunkable-and-intermediate subcategory* $\mathcal{M}_{lti}$.
  - The *negative-linear subcategory* $\mathcal{M}_l^{\ominus}$.
  - The *positive-thunkable subcategory* $\mathcal{M}_t^{\oplus}$.

Remark 3.24. 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.

Notation 3.25. We may write the sub- and super-scripts from Definition 3.23 onto the binary operators of morphisms ($\to$) or isomorphisms ($\cong$) to denote that the morphism or isomorphism (and its inverse) resides in the corresponding subcategory of the unital magmoid.

Convention 3.26. Phrasing such as "linear-and-thunkable isomorphism" has the meaning suggested by Notation 3.25. That is, a linear-and-thunkable isomorphism $f : a \cong_{lt} b$ has that both $f : a \to b$ and its inverse $f^{-1} : b \to a$ are both linear and thunkable.

3.3. Univalence

In this section we define and justify what it means for a unital magmoid to be *univalent*. Definition 3.28 defines a sort of "internal" univalence condition that describes identifications of objects **within** the unital magmoid. We also state and prove an "external" type of univalence that describes identifications **between** unital magmoids: that identifications between unital magmoids correspond to isomorphisms (Definition 3.32), and identifications between **univalent** magmoids correspond to weak equivalences of unital magmoids (Definition 3.36).

★ **Lemma 3.27.** *In a unital magmoid, linear-and-thunkable-and-intermediate isomorphisms $a \cong_{lti} b$ correspond to the indiscernibilities $a \asymp b$ of Ahrens et al. [3].*

Proof. See Appendix B. □

★ **Definition 3.28** (`is_unital_magmoid_univalent`)**.** A unital magmoid $\mathcal{M}$ is *univalent* if the linear-and-thunkable-and-intermediate subcategory $\mathcal{M}_{lti}$ is univalent.

Remark 3.29. Since all paths in a category $\mathcal{C}$ associate, $\mathcal{C}_{lti}$ is isomorphic to $\mathcal{C}$. Thus Definition 3.28 corresponds to Definition 3.8 in categories (see Lemma 3.34 below).

Remark 3.30. 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). If univalence were defined in terms of linear-and-thunkable isomorphisms instead, a univalent unital magmoid would have that all linear-and-thunkable isomorphisms are intermediate. This is because every morphism arising from `id_to_iso` $a\ b\ p$ shares all properties with identities id_a, by identification induction on p, and identities are intermediate (Example 3.22).

◉ **Definition 3.31** (`fully_faithful`)**.** A functor $F : \mathcal{N} \to \mathcal{M}$ 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 3.32** (`catiso`)**.** A categorical functor $F : \mathcal{N} \to \mathcal{M}$ between unital magmoids is an *isomorphism*, written $F : \mathcal{N} \cong \mathcal{M}$, if it is fully faithful and an equivalence on objects.

Remark 3.33 (weq_catiso_precategory_data_path'). When unital magmoids or categories $\mathcal{N}$ and $\mathcal{M}$ are $\mathcal{U}$-small for a univalent universe $\mathcal{U}$, isomorphisms $\mathcal{N} \cong \mathcal{M}$ are equivalent to identifications $\mathcal{N} = \mathcal{M}$, regardless of univalence of $\mathcal{N}$ and $\mathcal{M}$. Following Remark 2.17, however, we derive the following directly.

Lemma 3.34 (weq_is_univalent_over_catiso)**.** *If $\mathcal{C} \cong \mathcal{D}$ is an isomorphism of categories, then $\mathcal{C}$ is univalent if and only if $\mathcal{D}$ is.*

Proof. Use Theorem 3.10. □

Definition 3.35 (is_lti_essentially_surjective)**.** A functor $F\colon \mathcal{N} \to \mathcal{M}$ between unital magmoids is *split essentially surjective* when for all objects $a : \mathcal{M}$ there is a chosen object $b : \mathcal{N}$ and linear-and-thunkable-and-intermediate isomorphism $Fb \cong_{lti} a$ in $\mathcal{M}$. The functor F is *essentially surjective* when the object and isomorphism only merely exists, rather than there being a chosen pair.

Definition 3.36. A categorical functor $F\colon \mathcal{N} \to \mathcal{M}$ between unital magmoids is a *weak equivalence* if it is fully faithful and essentially surjective, and a *strong equivalence* if it is fully faithful and **split** essentially surjective.

Lemma 3.37 (lti_iso_from_fully_faithful_functor_image)**.** *If $F\colon \mathcal{N} \to \mathcal{M}$ is a fully faithful functor and $e : Fa \cong_{lti} Fb$ is a linear-and-thunkable-and-intermediate isomorphism, then its reflection $F_1^{-1}(e)\colon a \to b$ is also a linear-and-thunkable-and-intermediate isomorphism.*

Proof (sketch). Show that fully-faithful functors reflect linearity, thunkability, intermediateness, and inverses. □

Lemma 3.38 (isaprop_lti_iso_fiber_from_fully_faithful)**.** *Let $\mathcal{N}$ and $\mathcal{M}$ be unital magmoids, with $\mathcal{N}$ univalent. If a functor $F\colon \mathcal{N} \to \mathcal{M}$ is fully faithful, then the type $\sum_{b:\mathcal{N}} Fb \cong_{lti} a$ is a proposition for all a.*[3]

Proof. Let $\langle b, Fb \overset{e}{\cong}_{lti} a\rangle$ and $\langle b', Fb' \overset{e'}{\cong}_{lti} a\rangle$ be two elements of $\sum_{b:\mathcal{N}} Fb \cong_{lti} a$. We must show them to be identified. First, we find $Fb \overset{e}{\cong}_{lti} a \overset{e'^{-1}}{\cong}_{lti} Fb'$, and reflect it using Lemma 3.37 to obtain $F_1^{-1}(e \mathbin{⨾} e'^{-1}) : b \cong_{lti} b'$. By univalence of $\mathcal{N}$ and Theorem 3.10, we have $\langle b', F_1^{-1}(e \mathbin{⨾} e'^{-1})\rangle =_{\{b\}^{\rightrightarrows}_{\mathcal{N}}} \langle b, \mathsf{id}_b\rangle$. We thus obtain $b' = b$ and—after identification induction on that—$F_1^{-1}(e \mathbin{⨾} e'^{-1}) = \mathsf{id}_b$. Finally, $e = e'$ follows from the functor laws of F, and e' being a linear isomorphism. □

[3]This can be considered as F being "essentially injective."

Lemma 3.39 (`is_catiso_from_weak_um_equiv`)**.** *All weak equivalences between* ***univalent*** *unital magmoids are isomorphisms.*

Proof (sketch). Combine univalence and Lemma 3.38. □

4. Duploids

4.1. Definitions and Properties

[TODO: put **something** here]

Definition 4.1 (preduploid [25, 6, 20]). 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 [25, 6, 20]). 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. The only data of the polarity shifts is the action on objects $\Uparrow/\Downarrow$ and the first morphism $\texttt{force} / \texttt{wrap}$, the rest—negativity of $\Uparrow a$, linearity and isomorphism of $\texttt{force}_a$, and duals—is a mere proposition: the *axioms of polarity shifts*. Note also that thunkability of each $\texttt{delay}_a$ and linearity of each $\texttt{unwrap}_a$ follows 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 (recall Convention 2.18). This differs from the original construction by Munch-Maccagnoni [25] in which preduploids were defined in terms of separate negative and positive categories with additional mixed composition operators. We 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.

Definition 4.6 (`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 4.7. The original two-sorted duploids of Munch-Maccagnoni, 2014 [25] are precisely the split duploids where $\Uparrow$/ `force` and $\Downarrow$/ `wrap` are trivial—that is, identities—for syntactically negative and syntactically positive objects respectively.

Remark 4.8 (`polarity_mapping_to_has_polarities`, `polarity_mapping_from_LEM`). Every split preduploid is evidently a single-sorted preduploid. Conversely, the law of excluded middle allows defining a polarity mapping for any **preduploid**. For example, as

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

This does not help for defining a polarity mapping for **duploids**, as it will not respect the polarity shifts in general.

Definition 4.9 (`chosen_negative_category`, and dual)**.** 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* $\mathcal{D}^{\ominus\omega}$ of syntactically negative objects.
- The *syntactically positive subcategory* $\mathcal{D}^{\oplus\omega}$ of 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, and dual [25]). *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

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

Proof. This is Proposition 13 of Munch-Maccagnoni, 2014 [25]. □

Lemma 4.11 (is_positive_of_lt_iso). *Let $p : a \cong_{lt} b$ be a linear-and-thunkable isomorphism in a unital magmoid. 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 all objects c, all morphisms $f\colon b \to c$ are linear. The path $b \xrightarrow{p^{-1}}_{lt} a \xrightarrow{p}_{lt} b \xrightarrow{f} c$ associates by thunkability of p^{-1}, so we have

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

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

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

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

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

A. *a is positive.*

B. $\mathtt{wrap}_a : a \to_t \Downarrow a$ *is linear.*

C. $\mathtt{wrap}_a$ *is a linear-and-thunkable isomorphism $a \cong_{lt} \Downarrow a$ with inverse* $\mathtt{unwrap}_a$.

Proof. The implication $(2/B) \Rightarrow (3/C)$ is immediate. We have $(3/C) \Rightarrow (1/A)$ by Lemma 4.11, and $(1/A) \Rightarrow (2/B)$ by definition of the polarities. □

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.*

★ **Definition 4.15** (`is_duploid_univalent`). A duploid or preduploid $\mathcal{D}$ is *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`). This coincides with univalence of $\mathcal{D}$ as a unital magmoid by Corollary 4.14.

★ **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 $\mathcal{M}_t^{\oplus} \hookrightarrow \mathcal{M}_{lt} \hookleftarrow \mathcal{M}_l^{\ominus}$. For the "if" direction, we apply Theorem 3.10 and show that for all $a : \mathcal{D}$ the type of fans $\{a\}_{\mathcal{D}}^{\rightrightarrows}$ is a proposition. By the preduploid property, assume that a is positive (or negative). Then fans

$$\{a\}_{\mathcal{D}}^{\rightrightarrows} \equiv \sum_{b:\mathcal{D}} a \cong_{lt} b \qquad \text{are equivalent to fans}$$

$$\{a\}_{\mathcal{D}_t^{\oplus}}^{\rightrightarrows} \equiv \sum_{b:\mathcal{D}_t^{\oplus}} a \cong_t^{\oplus} b \quad \left(\text{or } \{a\}_{\mathcal{D}_l^{\ominus}}^{\rightrightarrows}\right) \qquad \text{by Lemma 4.11.}$$

The latter type is a proposition by assumption. □

For split (pre)duploids, Definition 4.15 is incorrect in general: the polarity mapping gives a way to differentiate between some linear-and-thunkably isomorphic objects.

★ **Theorem 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 negative and semantically positive, then the underlying single-sorted duploid of $\mathcal{D}$ is not univalent.*

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

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

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. $\square$

★ **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 syntactically negative-linear subcategory $\mathcal{D}_l^{\ominus\omega}$ and and its syntactically positive-thunkable subcategory $\mathcal{D}_t^{\oplus\omega}$ are univalent.

Remark 4.20. Definition 4.19 coincides with that derived for two-sorted duploids by Ahrens et al. [3, Example 13.5].

★ **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 $\langle \Uparrow, \texttt{force} \rangle$ and $\langle \Uparrow', \texttt{force}' \rangle$ for a preduploid $\mathcal{D}$. Then for all objects a, there is a linear isomorphism

$$I \overset{\text{def}}{\equiv} \Uparrow a \overset{\texttt{force}_a}{\cong_l} a \overset{\texttt{delay}'_a}{\cong_l} \Uparrow' a$$

that is also thunkable by the polarities of $\Uparrow$ and $\Uparrow'$. Moreover, $\texttt{force}_a : \Uparrow a \to a$ transported across I, as $I^{-1} \mathbin{⨾} \texttt{force}_a : \Uparrow' a \to a$, is equal to $\texttt{force}'_a : \Uparrow' a \to a$:

$$\begin{aligned} I^{-1} \mathbin{⨾} \texttt{force}_a &= (\texttt{force}_a \mathbin{⨾} \texttt{delay}'_a)^{-1} \mathbin{⨾} \texttt{force}_a \\ &= \texttt{force}'_a \mathbin{⨾} \texttt{delay}_a \mathbin{⨾} \texttt{force}_a \\ &= \texttt{force}'_a. \end{aligned}$$

It follows from univalence of $\mathcal{D}$ and Theorem 3.10 that $\langle \Uparrow' a, I \rangle =_{\{\Uparrow a\}^{\Rightarrow}_{\mathcal{D}_{lt}}} \langle \Uparrow a, \mathsf{id}_{\Uparrow a} \rangle$, and hence $\langle \Uparrow, \mathtt{force} \rangle = \langle \Uparrow', \mathtt{force}' \rangle$. The case is analogous for positive shifts. $\square$

5. Duploids from Adjunctions

5.1. The Oblique Duploid

The *oblique duploid* is the name I have given to the construction of the duploid arising from an adjunction, as defined by Munch-Maccagnoni [25]. In this section, we shall see how the oblique duploid construction gives rise to a distinctly **split** duploid, that is therefore unable to be univalent as a single-sorted duploid in general.

Definition 5.1 (`adjunction` [2]). 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":

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

The adjunction itself may be written as any variant of

$$L \dashv R, \qquad L \dashv_{(\eta,\epsilon)} R, \quad \text{or} \quad L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}.$$

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:

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

"Natural" here means, for all $f\colon p \to q$, $g\colon Lq \to n$, and $h\colon n \to m$, that

$$\varphi_{p,m}\left(Lp \xrightarrow{Lf} Lq \xrightarrow{g} n \xrightarrow{h} m\right) \;=\; p \xrightarrow{f} q \xrightarrow{\varphi_{q,n}(g)} Rn \xrightarrow{Rh} Rm.$$

Explicitly, φ may be given in terms of η and ϵ 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 [17], we may call the morphisms on either side of φ (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

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

Remark 5.5 (oblique_compose_negative, and dual). For locally small $\mathcal{N}/\mathcal{P}$, $\mathcal{O}$ 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 φ.

Remark 5.6 (isweq_oblique_mor_negative, and dual). 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

$$\left(g^{\flat^{-1}}\right)^{\sharp} \overset{\text{def}}{\equiv} \varphi_{p,n}(g) : \mathcal{P}[\![p, Rn]\!] \qquad \text{for } g : \mathcal{N}[\![Lp, n]\!], \text{ and}$$
$$\left(h^{\sharp^{-1}}\right)^{\flat} \overset{\text{def}}{\equiv} \varphi_{p,n}^{-1}(h) : \mathcal{N}[\![Lp, n]\!] \qquad \text{for } h : \mathcal{P}[\![p, Rn]\!].$$

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 $f^{\flat}$ and $f^{\sharp}$ 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 reasoning, including in a proof assistant.

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

$$\pi^{\ominus}_{\mathcal{O}}(\iota^{\ominus}\, n) \overset{\text{def}}{\equiv} n \qquad\qquad \pi^{\oplus}_{\mathcal{O}}(\iota^{\ominus}\, n) \overset{\text{def}}{\equiv} Rn$$
$$\pi^{\ominus}_{\mathcal{O}}(\iota^{\oplus}\, p) \overset{\text{def}}{\equiv} Lp \qquad\qquad \pi^{\oplus}_{\mathcal{O}}(\iota^{\oplus}\, p) \overset{\text{def}}{\equiv} p.$$

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

Definition 5.9 (`oblique_negative_identity`, and dual)**.** 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 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 $\mathsf{id}_a \overset{\text{def}}{\equiv} \mathsf{cross}\, a$ 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 $f : a \to b$ and $g : b \to c$ be morphisms in $\mathcal{O}[L \dashv R]$. Define

$f \mathbin{⨟} g \colon a \to c$ by discrimination on b, as either

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

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 define, for $a \xrightarrow{f} b \xrightarrow{g} c$:

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

This is the actual definition of `oblique_compose` used in the formalization, and by the original construction of Munch-Maccagnoni [25].

Lemma 5.15 (`oblique_left_id`, and dual)**.** *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 $f \colon a \to b$ in $\mathcal{O}[L \dashv R]$. Let us show that $\mathsf{id}_a \mathbin{⨟} f = f$. Suppose $a \equiv \iota^\oplus p$; then we have $(\mathsf{cross}\, a \mathbin{⨟} f)^\flat \equiv \mathsf{cross}\, a^\flat \mathbin{⨟} (\mathsf{cross}\, a^\flat)^{-1} \mathbin{⨟} f^\flat = f^\flat$. Thus we have $\mathsf{cross}\, a \mathbin{⨟} f = f$ 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`, and dual)**.** *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 $(f \mathbin{⨟} (g \mathbin{⨟} h))^\sharp = ((f \mathbin{⨟} g) \mathbin{⨟} h)^\sharp$ by discriminating on c.[1] The other polarity is symmetric. □

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

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*

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

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

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

For the morphisms $\mathtt{force}_a$ and $\mathtt{delay}_a$, compute their types:

$$\begin{aligned} \mathtt{force}_a : \mathcal{O}[L \dashv R]⟦\Uparrow a, a⟧ &\equiv \mathcal{O}⟦\pi^{\oplus}(\Uparrow a), \pi^{\ominus}\, a⟧ \\ &\equiv \mathcal{O}⟦\pi^{\oplus}(\Uparrow a), \pi^{\ominus}(\Uparrow a)⟧ \end{aligned} \qquad \begin{aligned} \mathtt{delay}_a : \mathcal{O}[L \dashv R]⟦a, \Uparrow a⟧ &\equiv \mathcal{O}⟦\pi^{\oplus}\, a, \pi^{\ominus} \Uparrow a⟧ \\ &\equiv \mathcal{O}⟦\pi^{\oplus}\, a, \pi^{\ominus}\, a⟧. \end{aligned}$$

Give $\mathtt{force}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}(\Uparrow a) : \mathcal{O}⟦\pi^{\oplus}(\Uparrow a), \pi^{\ominus}(\Uparrow a)⟧$ and $\mathtt{delay}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}\, a : \mathcal{O}⟦\pi^{\oplus}\, a, \pi^{\ominus}\, a⟧$.

Dually, define $\mathtt{wrap}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}(\Downarrow a)$ of the type $\mathcal{O}[L \dashv R]⟦a, \Downarrow a⟧ \equiv \mathcal{O}⟦\pi^{\oplus}(\Downarrow a), \pi^{\ominus}(\Downarrow a)⟧$, and $\mathtt{unwrap}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}\, a$ 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, and dual). *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, 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 [26]. □

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, and dual [25]). *Fix an adjunction $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$. A morphism $f\colon a \to \iota^{\oplus}_{\mathcal{O}}\, p$ in the oblique duploid*

$\mathcal{O}[L \dashv R]$ is thunkable if and only if the equation

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

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

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

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

$$(f \,⨾\, (\mathtt{delay}_{\iota^\oplus_{\mathcal{O}} p} \,⨾\, \mathtt{wrap}_{\Uparrow(\iota^\oplus_{\mathcal{O}} p)}))^\sharp = f^\sharp \,⨾\, RL\eta_p \qquad \text{on the left, and}$$

$$((f \,⨾\, \mathtt{delay}_{\iota^\oplus_{\mathcal{O}} p}) \,⨾\, \mathtt{wrap}_{\Uparrow(\iota^\oplus_{\mathcal{O}} p)})^\sharp = f^\sharp \,⨾\, \eta_{RLp} \qquad \text{on the right.}$$

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

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

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

Dually, for $n : \mathcal{N}$, the object $\iota^\ominus_{\mathcal{O}} n$ is semantically 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 Theorem 4.18 that the oblique duploid is not univalent as a single-sorted duploid.

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.

Example 5.24. Consider a fixed inhabited set s : **Set**, and the state adjunction $(-\times s) \dashv (s \Rightarrow -) : \mathbf{Set} \to \mathbf{Set}$, given by the isomorphism $\mathbf{Set}[\![a \times s, b]\!] \simeq \mathbf{Set}[\![a, s \Rightarrow b]\!]$ natural in $a : \mathbf{Set}^{\mathsf{op}}$ and $b : \mathbf{Set}$. The object $\iota^{\oplus}_{\mathcal{O}}\,\texttt{empty}$ (and $\iota^{\ominus}_{\mathcal{O}}\,\texttt{empty}$) is both semantically negative and positive in $\mathcal{O}[(- \times s) \dashv (s \Rightarrow -)]$.

5.2. The Envelope Duploid

To define a truly single-sorted duploid arising from an adjunction, we must handle with care the objects that are both semantically positive and negative. For it to be univalent, we must (by Theorem 4.18) ensure that no polarity mapping can be defined when such an object exists. As anticipated in Remark 5.14, we obtain a perfectly serviceable version of the oblique duploid by treating the cross morphisms as merely isomorphisms in one of the categories. Crucially, we forget—by propositional truncation—which category it is. In this section, we give the construction of the novel *envelope duploid*, a single-sorted counterpart to the oblique duploid.

Definition 5.25 (`envelope_preob`)**.** Let $L \dashv R : \mathcal{N} \to \mathcal{P}$ be an adjunction. An *envelope pre-object* a in $L \dashv R$ is a triple of a positive object $\pi^{\oplus}_{\mathcal{E}}\,a : \mathcal{P}$, a negative object $\pi^{\ominus}_{\mathcal{E}}\,a : \mathcal{N}$, and an oblique morphism $\mathsf{cross}_{\mathcal{E}}\,a : \mathcal{O}_{L\dashv R}[\![p, n]\!]$ between them.

Notation 5.26. We may write an 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`, and dual)**.** 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 $\mathcal{O}$-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

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

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

Lemma 5.30 (envelope_chosen_negative_of_negative, and dual)**.** *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. Compute $\mathsf{cross}_{\mathcal{O}}(\iota^{\ominus}_{\mathcal{O}}\, n)^{\sharp} \equiv \mathsf{id}_{Rn}$ and $\mathsf{cross}_{\mathcal{O}}(\iota^{\oplus}_{\mathcal{O}}\, p)^{\flat} \equiv \mathsf{id}_{Lp}$. □

Definition 5.31 (envelope_ob, envelope_mor)**.** Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The *envelope duploid-to-be* $\mathcal{E}[L \dashv R]$ over $L \dashv R$ has objects, morphisms, and identities 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 $\mathsf{id}_a \stackrel{\text{def}}{\equiv} \mathsf{cross}\, a$.

Remark 5.32. Beware Convention 2.18: 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 5.33 (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

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

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

Remark 5.34. This is identical to Definition 5.13 of composition in the oblique duploid, but on envelope pre-objects instead.

Theorem 5.35 (`envelope_compose_irrel`)**.** *Let $f: a \to b$ and $g : b \to c$ be morphisms in the envelope duploid-to-be $\mathcal{E}[L \dashv R]$, with there being both a negative and a positive polarization choice for b. Then the two ways of composing $f \mathbin{⨾} g: a \to c$ using Definition 5.33 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:

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

In fact, we need only check that $L(\mathsf{cross}\, b^\sharp)^{-1}$ and $R(\mathsf{cross}\, b^\flat)^{-1}$ are mates, since it follows by naturality of φ that

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

Therefore we must show $R(\mathsf{cross}\, b^\flat)^{-1} = \varphi_{R(\pi^\ominus b),L(\pi^\oplus b)}(L(\mathsf{cross}\, b^\sharp)^{-1})$. By isomorphism of $\mathsf{cross}\, b^\sharp$ and φ, it suffices to consider

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

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

★ **Definition 5.36** (`envelope_compose`, `envelope_unital_magmoid`). The envelope duploid-to-be $\mathcal{E}[L \dashv R]$ has composition determined by Definition 5.33 from the polarization choice that exists for each envelope object. By Theorem 5.35 the choice of polarization is irrelevant, thus this definition is unique. Identities are given by $\mathsf{id}_a \overset{\text{def}}{\equiv} \mathsf{cross}_{\mathcal{E}}\, a$.

★ **Lemma 5.37** (`is_negative_of_envelope_chosen_negative`, and dual). *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. □

★ **Lemma 5.38** (`envelope_has_negative_shifts`, and dual). *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,*

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

Proof. Same as Lemma 5.19. □

★ **Corollary 5.39** (`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.

★ **Lemma 5.40** (`weak_equiv_oblique_to_envelope_duploid`). *The oblique duploid $\mathcal{O}[L \dashv R]$ is weakly equivalent (in the sense of Definition 3.36) to the corresponding envelope duploid $\mathcal{E}[L \dashv R]$.*

Proof (sketch). The 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 $F_1\, f \overset{\text{def}}{\equiv} f$. Being fully-faithful is immediate; the pre-image for objects is given by choosing for each $a : \mathcal{E}[L \dashv R]$ (under propositional truncation) either $\iota^{\ominus}_{\mathcal{O}}(\pi^{\ominus}_{\mathcal{E}}\, a)$ or $\iota^{\oplus}_{\mathcal{O}}(\pi^{\oplus}_{\mathcal{E}}\, a)$ based on the polarization choice of a. □

Remark 5.41 (`equiv_oblique_to_envelope_duploid_from_LEM`). The equivalence of Lemma 5.40 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 $\iota^{\ominus}_{\mathcal{O}}(\pi^{\ominus}_{\mathcal{E}}\, a)$ when it does.

Lemma 5.42 (is_thunkable_from_downshift_iff_envelope_unit_postcompose, and dual [25]). *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*

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

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

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

Proof. Same as Lemma 5.21. □

Lemma 5.43 (is_negative_envelope_downshift_iff_pre_fixed_point, and dual). *Let $L \dashv_{(\eta,\epsilon)} R : \mathcal{N} \to \mathcal{P}$ be an adjunction. For all $p : \mathcal{P}$, 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 $n : \mathcal{N}$, 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 5.42. □

5.3. Univalence and the Equalizing Requirement

Under what conditions are the oblique and envelope duploids of an adjunction $L \dashv R : \mathcal{P} \to \mathcal{N}$ univalent? It seems reasonable to assume that $\mathcal{N}$ and $\mathcal{P}$ must be univalent. We must somehow show that $\mathcal{O}[L \dashv R]^{\ominus\omega}_{l}$ and $\mathcal{O}[L \dashv R]^{\oplus\omega}_{t}$ (Definition 4.19), or $\mathcal{E}[L \dashv R]^{\ominus}_{l}$ and $\mathcal{E}[L \dashv R]^{\oplus}_{t}$ (Theorem 4.17), are univalent. We shall do so using Lemma 3.34, by exhibiting isomorphisms of categories with $\mathcal{N}$ and $\mathcal{P}$ respectively. To do so, we will need what Munch-Maccagnoni called the *equalizing requirement* (Definition 5.50) [25].

Lemma 5.44 (negative_to_oblique_duploid, negative_category_to_envelope_duploid, etc.). *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:*

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

These are given on morphisms by

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

Proof. The respective linearity and thunkability conditions follow from Lemma 5.42 and naturality of ϵ/η. The functor laws follow from those of L and R. □

Lemma 5.45 (`isweq_on_objects_negative_to_oblique_duploid`, and dual)**.** *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 5.46 (`envelope_ob_eq_negative'`, and dual)**.** *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 former. We need to identify

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

By assumption, $\mathsf{cross}\, a^{\sharp} : \mathcal{P}[\![\pi^{\oplus}\, a, R(\pi^{\ominus}\, a)]\!]$ is an isomorphism. Thus by Theorem 3.10 we have $\langle \pi^{\oplus}\, a, \mathsf{cross}\, a^{\sharp}\rangle =_{\{R(\pi^{\ominus}\, a)\}^{\leftarrow}_{\mathcal{P}}} \langle R(\pi^{\ominus}\, a), \mathsf{id}_{R(\pi^{\ominus}\, a)}\rangle$. Since $\mathsf{id}_{R(\pi^{\ominus}\, a)} \equiv \mathsf{cross}(\Uparrow a)^{\sharp}$, Equation 5.4 follows from injectivity of $\sharp$. □

Remark 5.47. Unlike for the oblique duploid, Lemma 5.46 is insufficient to make $\iota^{\ominus}_{\mathcal{E}}$ and $\iota^{\oplus}_{\mathcal{E}}$ equivalences on objects. This is because, say, a semantically negative object $\iota^{\oplus}_{\mathcal{E}}\, p : \mathcal{E}[L \dashv_{(\eta,\epsilon)} R]^{\ominus}$ for some $p : \mathcal{P}$ need only have $RL\eta_p = \eta_{RLp}$ (Lemma 5.43), which does not necessarily imply that η_p has an inverse. The equalizing requirements of Definition 5.50 below will remedy this.

Lemma 5.48 (split_essentially_surjective_negative_category_to_envelope_duploid). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. 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 essentially surjective.*

Proof. The pre-image of $b : \mathcal{E}[L \dashv R]^{\ominus}_{l}$ in $\iota^{\ominus}_{\mathcal{E}}$ is $\pi^{\ominus}_{\mathcal{E}}\, b : \mathcal{N}$. We have $\iota^{\ominus}_{\mathcal{E}}(\pi^{\ominus}_{\mathcal{E}}\, b) \equiv \Uparrow b$, so the isomorphism in $\mathcal{E}[L \dashv R]^{\ominus}_{l}$ is just $\Uparrow b \overset{\texttt{force}_b}{\cong_{lt}} b$. The case is dual for $\iota^{\oplus}_{\mathcal{E}}$. □

Theorem 5.49 (fully_faithful_negative_category_to_envelope_duploid_iff, etc.). *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. The following are equivalent:*

1. *The functor $\iota^{\ominus}_{\mathcal{O}} \colon \mathcal{N} \to \mathcal{O}[L \dashv R]^{\ominus\omega}_{l}$ is fully-faithful.*
2. ——————————————— *is an isomorphism of categories.*
3. *The functor $\iota^{\ominus}_{\mathcal{E}} \colon \mathcal{N} \to \mathcal{E}[L \dashv R]^{\ominus}_{l}$ is fully-faithful.*
4. ——————————————— *is a strong equivalence of categories.*[2]
5. *For all $n : N$, the fork $LRLRn \;\substack{\xrightarrow{LR\epsilon_n} \\ \xrightarrow[\epsilon_{LRn}]{}}\; LRn \xrightarrow{\epsilon_n} n$ 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 $f \colon LRn \to m$ satisfying $LR\epsilon_n \mathbin{⨟} f = \epsilon_{LRn} \mathbin{⨟} f$, there is a unique morphism $f^* \colon n \to m$ such that $\epsilon_n \mathbin{⨟} f^* = f$.

Dually, the following are equivalent:

A. *The functor $\iota^{\oplus}_{\mathcal{O}} \colon \mathcal{N} \to \mathcal{O}[L \dashv R]^{\oplus\omega}_{t}$ is fully-faithful.*

B. ——————————————— *is an isomorphism of categories.*

C. *The functor $\iota^{\oplus}_{\mathcal{E}} \colon \mathcal{N} \to \mathcal{E}[L \dashv R]^{\oplus}_{t}$ is fully-faithful.*

D. ——————————————— *is a strong equivalence of categories.*[2]

[2] Conditions 4 and D are not propositions in general, so "equivalent" means "if and only if" for these.

E. 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 $g\colon p \to RLq$ satisfying $g \mathbin{⨟} RL\eta_q = g \mathbin{⨟} \eta_{RLq}$, there is a unique morphism $g^*\colon p \to q$ such that $g^* \mathbin{⨟} \eta_q = g$.

Proof. The implications (2/B) $\Rightarrow$ (1/A) and (4/D) $\Rightarrow$ (3/C) are immediate. Their converses (1/A) $\Rightarrow$ (2/B) and (3/C) $\Rightarrow$ (4/D) follow from Lemma 5.45 and Lemma 5.48. The equivalences (1/A) $\Leftrightarrow$ (3/C) $\Leftrightarrow$ (5/E) follow from the characterizations of linear and thunkable morphisms in terms of the counit ϵ and unit η of the adjunction (Lemma 5.21 and Lemma 5.42). □

Definition 5.50 (`is_negative_equalizing`, `is_fully_equalizing`, and dual [25])**.** We say that an adjunction satisfying conditions 1–5 of Theorem 5.49 is *negative equalizing*; and that an adjunction satisfying conditions A–E is *positive equalizing*. We say that an adjunction which satisfies conditions 1–5 **and** conditions A–E is *fully equalizing*.

Remark 5.51. By other authors [4, 19], 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 5.52 (`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 $\mathcal{N}$ and $\mathcal{P}$ are univalent.*

Proof. Recall that univalence as a split duploid means that $\mathcal{O}[L \dashv R]_l^{\ominus\omega}$ and $\mathcal{O}[L \dashv R]_t^{\oplus\omega}$ are univalent. Using Lemma 3.34, univalence may be transported (in both directions) across the isomorphisms $\iota_{\mathcal{O}}^{\ominus} : \mathcal{N} \cong \mathcal{O}[L \dashv R]_l^{\ominus\omega}$ and $\iota_{\mathcal{O}}^{\oplus} : \mathcal{P} \cong \mathcal{O}[L \dashv R]_t^{\oplus\omega}$. □

Lemma 5.53 (`is_negative_fixed_point_iff_pre_fixed_point`, and dual)**.** *Let $L \dashv R : \mathcal{P} \to \mathcal{N}$ be an adjunction. If $L \dashv R$ is negative equalizing, then for all objects*

$n : \mathcal{N}$,

$$LR\epsilon_n = \epsilon_{LRn} \quad \textit{implies that } \epsilon_n \textit{ is an isomorphism.}$$

Dually, if $L \dashv R$ *is positive equalizing, then for all objects* $p : \mathcal{N}$,

$$RL\eta_p = \eta_{RLp} \quad \textit{implies that } \eta_p \textit{ is an isomorphism.}$$

Proof (sketch). This follows from the coequalizer and equalizer conditions of Theorem 5.49, instantiated with $f \equiv \mathsf{id}_{LRn}$ and $g \equiv \mathsf{id}_{RLp}$ respectively. □

★ **Lemma 5.54** (`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, and let $\varphi_{p,n} : \mathcal{N}[\![Lp, n]\!] \simeq \mathcal{P}[\![p, Rn]\!]$ be the natural isomorphism witnessing $L \dashv R$. 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, so $\mathsf{cross}\, a^{\flat}$ is an isomorphism. We show that $\mathsf{cross}\, a^{\sharp} = \varphi_{(\pi^{\oplus} a),(\pi^{\ominus} a)}(\mathsf{cross}\, a^{\flat}) \equiv \eta_{\pi^{\oplus} a} \mathbin{⨾} R(\mathsf{cross}\, a^{\flat})$ is the composite of two isomorphisms. $R(\mathsf{cross}\, a^{\flat})$ being an isomorphism follows from our assumption. Finally, $\eta_{\pi^{\oplus} a}$ being an isomorphism follows from Lemma 5.53 and Lemma 5.42; since $\Downarrow a$, being linearly-and-thunkably isomorphic to a, is positive. □

★ **Lemma 5.55** (`is_catiso_negative_category_to_envelope_duploid`, and dual)**.** *Let* $L \dashv R : \mathcal{P} \to \mathcal{N}$ *be a fully equalizing adjunction between* ***univalent*** *categories* $\mathcal{P}$ *and* $\mathcal{N}$*. The inclusions* $\iota^{\ominus}_{\mathcal{E}} : \mathcal{N} \to \mathcal{E}[L \dashv R]^{\ominus}_{l}$ *and* $\iota^{\oplus}_{\mathcal{E}} : \mathcal{P} \to \mathcal{E}[L \dashv R]^{\oplus}_{t}$ *are isomorphisms of categories.*

Proof. Combine Lemma 5.54 and Lemma 5.46. □

★ **Theorem 5.56** (`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 5.52, using Theorem 4.17 and Lemma 5.55. □

Remark 5.57. As per Remark 2.17, Theorem 5.56 does not use the univalence axiom.

6. Conclusions and Future Work

- Contributions
  - Formalized single-sorted and split duploids.
  - Defined univalence conditions.
  - Constructed two duploids arising from an adjunction, one univalent as split, one univalent as a single-sorted.
- Future work
  - Rezk completion. We know how this could be done, but have not managed to formalize it.
  - Study of the bicategory of duploids, including full structure theorem for two-sorted duploids (i.e. reflection in the bicategory of adjunctions). We did not manage to formalize these either.
  - Study of structures on univalent duploids: universal properties in general and their uniqueness.

Bibliography

[1] B. Ahrens, D. Frumin, M. Maggesi, N. Veltri, and N. van der Weide, "Bicategories in univalent foundations," *Mathematical Structures in Computer Science*, vol. 31, no. 10, pp. 1232–1269, 2021. DOI: 10.1017/S0960129522000032

[2] 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

[3] 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.

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

[5] A. Bordg, *Univalent foundations and the unimath library*, 2019. DOI: 10.1007/978-3-030-15655-8_8 arXiv: 1710.02723 [math.LO]. [Online]. Available: https://arxiv.org/abs/1710.02723

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] D. R. Grayson et al., *UniMath/UniMath*, version v20250923, Sep. 2025. DOI: 10.5281/zenodo.17186647

[12] 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

[13] M. Hofmann and T. Streicher, "The groupoid interpretation of type theory," in *Twenty-five years of constructive type theory (Venice, 1995)*, ser. Oxford Logic Guides, vol. 36, New York: Oxford Univ. Press, 1998, pp. 83–111.

[14] 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

[15] 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

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

[17] 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

[18] 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.

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

[20] 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

[21] 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

[22] 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

[23] 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

[24] 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

[25] 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

[26] 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

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

[28] The Rocq Development Team. "The Rocq prover 9.1.1 documentation. "[Online]. Available: https://rocq-prover.org/doc/v9.1/refman/index.html

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

[30] N. van der Weide, “The formal theory of monads, univalently,” *Logical Methods in Computer Science*, vol. Volume 21, Issue 1, Feb. 2025, ISSN: 1860-5974. DOI: 10.46298/lmcs-21(1:16)2025

[31] K. Wullaert, R. Matthes, and B. Ahrens, “Univalent monoidal categories,” en, vol. 269, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 15:1–15:21. DOI: 10.4230/LIPICS.TYPES.2022.15 [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TYPES.2022.15

Index

A. Counterexample Unital Magmoids

This section includes some examples of unital magmoids, to show that certain properties obtained in categories and preduploids do not generalize to unital magmoids.

★ **Example A.1** (lini)**.** There is a unital magmoid **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 : **Lini** and the morphisms

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

such that $p \mathbin{⨾} q = \mathsf{id}_b$ and $q \mathbin{⨾} p = \mathsf{id}_a$. 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

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

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 $[f\,;g]$: $\mathbf{Lini}[\![a, c]\!]$ is just notation for, say `false` : `bool`. Composition is uniquely determined by unitality and the equations

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

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

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

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

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

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.

B. Yoneda and Indiscernibilities in Unital Magmoids

In this section, we prove Lemma 3.27. To do so, we prove a slight generalization of part of Yoneda's lemma.

Lemma 3.27 (restatement from page 24)**.** *In a unital magmoid, linear-and-thunkable-and-intermediate isomorphisms $a \cong_{lti} b$ correspond to the indiscernibilities $a \asymp b$ of Ahrens et al. [3].*

Lemma B.1 (`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

$$(f \mathbin{⨟} p) \mathbin{⨟} (p^{-1} \mathbin{⨟} g) = f \mathbin{⨟} (p \mathbin{⨟} p^{-1}) \mathbin{⨟} g = f \mathbin{⨟} g. \qquad \square$$

Lemma B.2 (`is_intermediate_from_interpose`)**.** *In a unital magmoid, if $p : b \cong b'$ is an isomorphism with thunkable inverse $p^{-1} : b' \to_t b$, and p satisfies $(f \mathbin{⨟} p) \mathbin{⨟} (p^{-1} \mathbin{⨟} g) = f \mathbin{⨟} g$ 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 $f \mathbin{⨟} (p \mathbin{⨟} g) = (f \mathbin{⨟} p) \mathbin{⨟} g$. We have that $b' \xrightarrow{p^{-1}}_t b \xrightarrow{p} b' \xrightarrow{g} c$ associates by thunkability of p^{-1}. Thus we have

$$f \mathbin{⨟} (p \mathbin{⨟} g) = (f \mathbin{⨟} p) \mathbin{⨟} (p^{-1} \mathbin{⨟} p \mathbin{⨟} g) = (f \mathbin{⨟} p) \mathbin{⨟} g \qquad \text{by assumption.} \qquad \square$$

Definition B.3 (`um_yoneda`)**.** Let $\mathcal{U}$ be a universe and $\mathcal{M}$ be a locally $\mathcal{U}$-small unital magmoid. There is a reflexive graph functor $\mathsf{y}[\mathcal{M}]\colon \mathcal{M} \dashrightarrow \mathbf{RxFn}[\mathcal{M}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}]$, or simply

y, given by

$$\begin{aligned} \mathsf{y}_0(a)_0(b) &\stackrel{\text{def}}{\equiv} \mathcal{M}[\![b,a]\!] \\ \mathsf{y}_0(a)_1(c \xrightarrow{g} b)(b \xrightarrow{h} a) &\stackrel{\text{def}}{\equiv} c \xrightarrow{g} b \xrightarrow{h} a \\ \mathsf{y}_1(a \xrightarrow{g} b)_c(c \xrightarrow{f} a) &\stackrel{\text{def}}{\equiv} c \xrightarrow{f} a \xrightarrow{g} b. \end{aligned}$$

Remark B.4. When $\mathcal{M}$ is a category, y becomes a functor $\mathsf{y}\colon \mathcal{M} \to \mathbf{Fn}[\mathcal{M}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}]$ and satisfies an isomorphism

$$\mathbf{Fn}[\mathcal{M}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}][\![\mathsf{y}(a), F]\!] \simeq Fa$$
$$\text{natural in } F\colon \mathbf{Fn}[\mathcal{M}^{\mathsf{op}}, \mathbf{Set}_{\mathcal{U}}] \text{ and } a : \mathcal{M}^{\mathsf{op}}.$$

This is *Yoneda's lemma* for categories. The reader is encouraged to see which parts of the proof fail in the absence of associativity.

Lemma B.5 (`um_yoneda_linear`)**.** *When restricted to linear maps,* y *becomes a categorical functor, and transformations* $\mathsf{y}_1(f\colon a \to_l b)$ *are natural.*

Proof. Immediate from unfolding the definitions. □

Lemma B.6 (`um_yoneda_linear_fully_faithful`)**.** *The restricted and co-restricted functor* $\mathsf{y}_l\colon \mathcal{M}_l \to \mathbf{RxFn}_{\mathsf{nat}}[\mathcal{M}^{\mathsf{op}}, \mathbf{Set}]$ *is fully-faithful.*

Proof. The inverse takes $\alpha\colon \mathsf{y}_0(a) \Rightarrow \mathsf{y}_0(b)$ to $\alpha_a(\mathsf{id}_a) : \mathcal{M}[\![a,b]\!]$, which is linear by naturality of α. □

Corollary B.7 (`weq_um_yoneda_linear_isos`)**.** *Let* $\mathcal{M}$ *be a unital magmoid. Linear isomorphisms* $a \cong_l b$ *in* $\mathcal{M}$ *are equivalent to natural isomorphisms* $\mathsf{y}[\mathcal{M}](a) \cong \mathsf{y}[\mathcal{M}](b)$*. Dually, thunkable isomorphisms* $a \cong_t b$ *are equivalent to natural isomorphisms* $\mathsf{y}[\mathcal{M}^{\mathsf{op}}](a) \cong \mathsf{y}[\mathcal{M}^{\mathsf{op}}](b)$*.*

Proof of Lemma 3.27. Our diagram signature is exactly the same as categories, but we do not impose the associativity axiom. Therefore most of the extended example in Chapter 3 of Ahrens et al. [3] applies. One can verify that the data and properties (3.13–3.26) of indiscernibilities in Section 3.4 correspond to a linear-and-thunkable-and-intermediate isomorphism. This follows from the previous lemmas (B.1, B.2, B.6).

Let $\mathcal{M}$ be a unital magmoid. The data of an indiscernibility [3] $a \asymp b$ includes a natural isomorphism in the codomain (3.13, 3.16) $\mathsf{y}[\mathcal{M}](a) \cong \mathsf{y}[\mathcal{M}](b)$, hence a linear isomorphism $p : a \cong_l b$; and a natural isomorphism in the domain (3.14, 3.18) $\mathsf{y}[\mathcal{M}^{\mathsf{op}}](a) \cong \mathsf{y}[\mathcal{M}^{\mathsf{op}}](b)$, hence a thunkable isomorphism $q : a \cong_t b$. These are forced to be the same by the equation $(f \mathbin{;} p) \mathbin{;} (q^{-1} \mathbin{;} g) = f \mathbin{;} g$ for all $x \xrightarrow{f} a \xrightarrow{g} z$ (3.17), and the same property also forces the isomorphism to be intermediate. Thus we have $p = q : a \cong_{lti} b$. The last bit of data is an identity-preserving isomorphism (3.15, 3.23) $\mathcal{M}[\![a, a]\!] \cong \mathcal{M}[\![b, b]\!]$ which is forced to be $f \mapsto p^{-1} \mathbin{;} f \mathbin{;} p$ (3.19–3.21). The remaining properties (3.22–3.23, 3.24–3.26) are automatic. $\square$