Models of a Non-Associative Composition[⋆]

Guillaume Munch-Maccagnoni

Univ Paris Diderot, Sorbonne Paris Cité, PPS, UMR 7126 CNRS,
PiR2, INRIA Paris-Rocquencourt, F-75205 Paris, France.

Abstract We characterise the polarised evaluation order through a categorical structure where the hypothesis that composition is associative is relaxed. Duploid is the name of the structure, as a reference to Jean-Louis Loday's duplicial algebras. The main result is a reflection $\mathcal{A}\mathbf{dj} \to \mathcal{D}\mathbf{upl}$ where $\mathcal{D}\mathbf{upl}$ is a category of duploids and duploid functors, and $\mathcal{A}\mathbf{dj}$ is the category of adjunctions and pseudo maps of adjunctions. The result suggests that the various biases in denotational semantics: indirect, call-by-value, call-by-name... are ways of hiding the fact that composition is not always associative.

1 Introduction

In a term language where the order of evaluation is determined by the polarity of a type or formula, it is not immediate that composition is associative. The associativity of categorical composition amounts to the following equation on terms:

$$\mathsf{let}\ y\ \mathsf{be}\ t\ \mathsf{in}\ \mathsf{let}\ x\ \mathsf{be}\ u\ \mathsf{in}\ v \overset{?}{=} \mathsf{let}\ x\ \mathsf{be}\ (\mathsf{let}\ y\ \mathsf{be}\ t\ \mathsf{in}\ u)\ \mathsf{in}\ v$$

Now, in any setting where t could be of a type that implies a strict evaluation order, and where u could be of a type that implies a delayed evaluation, we can see a difference in spirit between these two terms. Indeed, in the left-hand term, the evaluation of t would happen before the one of v. On the contrary, in the right-hand term, the evaluation of x is delayed since it has the same type as u, so the term would compute v before t.

This phenomenon is observed with polarisation in logic and denotational semantics [7,21,5,16,12]. Polarisation can thus be described (negatively) as rejecting, either directly or indirectly, the hypothesis that composition is *a priori* associative. In this article, we give a positive and direct description of a polarised evaluation order. To this effect we introduce a category-like structure where not all composites associate. *Duploid* is the name of the structure, as a reference to Jean-Louis Loday's duplicial algebras [15].

The main result relates duploids to adjunctions. To help understand this relation, let us first recall the correspondence between direct models of call by value and indirect models *à la* Moggi.

Direct models. In a *direct* denotational model, there should be a close match between the given operations in the model and the constructions in the language. Essentially, type and program constructors should respectively correspond to operations on objects

⋆ This is a shortened version of Chapter II from the author's PhD thesis [20, pp. 86-91,103-152].

Table 1: Comparison of the structures underlying various direct models of computation.

Evaluation order	By value	By name	Polarised
Direct model	Thunk	Runnable monad	Duploid
Indirect model	Monad T	Co-monad L	Adjunction $F \dashv G$
Programs	Kleisli maps $P \to TQ$	Co-Kleisli maps $LN \to M$	Oblique maps $FP \to N$ $\simeq$ $P \to GN$
Syntactic data	Values	Stacks	Both
Completion into	Thunkable expressions	Linear evaluation contexts	Both

and on morphisms in a category. In particular, it should be possible to reason about an instance of the model within the language.[1] An example of direct models for the simply-typed lambda calculus is given by cartesian-closed categories.

In a model such as Moggi's λ_C models [18], or Lafont, Reus and Streicher's models of call by name [10], however, the language is not interpreted directly but through a Kleisli construction for a monad or a co-monad. We have a precise description of the link between direct models and indirect models thanks to Führmann [6]. Categories that model call by value directly are characterised by the presence of a *thunk*, a formal account of the well-known structure used to implement laziness in call-by-value languages.

The characterisation takes the following form: any direct model arises from the Kleisli construction starting from a λ_C model. However, from the direct model we can only recover a specific λ_C model: its values are made of all the pure expressions. More precisely, the Kleisli construction is a *reflection* that conflates any two values equalised by the monad, and turns into a value any *thunkable* expression. An expression is thunkable if it behaves similarly to a value in a sense determined by the monad.

Selinger [23] proves a similar relationship between direct models of the call-by-name $\lambda\mu$ calculus and Lafont, Reus and Streicher's models [10].

Adjunction-based models. This article deals with the underlying algebraic structure in these models: a monad over a category of values for call by value, a co-monad over a category of stacks for call by name. Duploids generalise the underlying structure to an adjunction between a category of values and a category of stacks. (See table 1.)

Relationship with polarities comes from Girard's *polarised translation* of classical logic [7,5,11]. Our duploid construction extends the (skeleton of the) polarised translation to any adjunction. (Notably, we do not need the assumption that there is an involutive negation operation on formulae.)

We know that there is a practical relevance of decomposing monads, when seen as notions of computation, into adjunctions, thanks to Levy [13,14]. Levy's adjunctions subsume models of call by value and call by name. However the model is indirect, and still lacks a corresponding notion of direct model.

[1] Führmann [6], Selinger [22].

Outline. Section 2 introduces pre-duploids as categories where the associativity of composition is deficient. Section 3 defines duploids as pre-duploids with additional structure, and characterises this additional structure. The category $\mathcal{D}upl$ of duploids and duploid functors is introduced. Section 4 proves the main result.

Structure theorem. The main result is a reflection $\boxed{\mathcal{D}upl \triangleleft \mathcal{A}dj}$, where $\mathcal{A}dj$ is the category of adjunctions and *pseudo maps of adjunctions*. In other words, the duploid construction extends to a functor $\mathcal{A}dj \to \mathcal{D}upl$ that admits a full and faithful right adjoint. In particular, any duploid is obtained from an adjunction, but adjunctions obtained from duploids are peculiar.

As a consequence of the main result, duploids account for a wide range of computational models, as we will see in various examples. It suggests that the various biases in denotational semantics: indirect, call-by-value, call-by-name… are ways of hiding the fact that composition is not always associative.

In addition, the article develops an internal language for duploids. It provides intuitions from programming languages and abstract machines about polarisation.

Characterisation of duploids. We also characterise the adjunctions obtained from duploids. We show that there is an equivalence of categories $\boxed{\mathcal{D}upl \simeq \mathcal{A}dj_{\mathbf{eq}}}$, where $\mathcal{A}dj_{\mathbf{eq}}$ is the full subcategory of adjunctions that satisfies the *equalizing requirement*: the unit and the co-unit of the adjunction are respectively equalisers and co-equalisers.

This means that the duploid operates from the point of view of the model of computation defined by the adjunction: first any two values and any two stacks that are not distinguished by the model of computation are identified; and then the categories of values and stacks are respectively completed with all the expressions that are thunkable, and with all the evaluation contexts that are linear.

2 Pre-duploids

We define pre-duploids, which are category-like structures whose objects have a polarity, and which miss associativity of composition when the middle map has polarity $+ \to \ominus$.

Definition 1. *A* pre-duploid $\mathscr{D}$ *is given by:*

1. *A set* $|\mathscr{D}|$ *of* objects *together with a* polarity mapping $\varpi : |\mathscr{D}| \to \{+,\ominus\}$.
2. *For all* $A, B \in |\mathscr{D}|$, *a set of* morphisms *or* hom-set $\mathscr{D}(A,B)$.
3. *For all morphisms* $f \in \mathscr{D}(A,B)$ *and* $g \in \mathscr{D}(B,C)$, *a morphism* $g \circledcirc f \in \mathscr{D}(A,C)$, *also written as follows depending on the polarity of* B*:*

$$g \bullet f \in \mathscr{D}(A,C) \text{ if } \varpi(B) = +\,,$$
$$g \circ f \in \mathscr{D}(A,C) \text{ if } \varpi(B) = \ominus\,.$$

The following associativities must hold for all objects $A, B \in |\mathscr{D}|$*;* $P, Q \in \varpi^{-1}(\{+\})$ *and* $N, M \in \varpi^{-1}(\{\ominus\})$*:*

(••) *For all* $A \xrightarrow{f} P \xrightarrow{g} Q \xrightarrow{h} B$, *one has* $(h \bullet g) \bullet f = h \bullet (g \bullet f)$;

($\circ\circ$) *For all* $A \xrightarrow{f} N \xrightarrow{g} M \xrightarrow{h} B$*, one has* $(h \circ g) \circ f = h \circ (g \circ f)$;

($\bullet\circ$) *For all* $A \xrightarrow{f} N \xrightarrow{g} P \xrightarrow{h} B$*, one has* $(h \bullet g) \circ f = h \bullet (g \circ f)$*.*

4. *For all* $A \in |\mathscr{D}|$*, a morphism* $\mathrm{id}_A \in \mathscr{D}(A,A)$ *neutral for* $\circledcirc$*.*

The mapping ϖ defines a partition of $|\mathscr{D}|$ into the *positive* objects $P, Q...$ in $|\mathscr{P}| \overset{\text{def}}{=} \varpi^{-1}(\{+\})$ and the *negative* objects $N, M...$ in $|\mathscr{N}| \overset{\text{def}}{=} \varpi^{-1}(\{\ominus\})$. This partition defines categories $\mathscr{P}$ (whose composition is given by $\bullet$) and $\mathscr{N}$ (whose composition is given by $\circ$) in an obvious way.

2.1 Linear and Thunkable Morphisms

Definition 2. *Let $\mathscr{D}$ be a pre-duploid. A morphism f of $\mathscr{D}$ is* linear *if for all g, h one has:*

$$f \circledcirc (g \circledcirc h) = (f \circledcirc g) \circledcirc h$$

A morphism f of $\mathscr{D}$ is thunkable *if for all g, h one has:*

$$h \circledcirc (g \circledcirc f) = (h \circledcirc g) \circledcirc f$$

Thus any morphism $f : P \to A$ is linear, and any morphism $f : A \to N$ is thunkable. The terminology *thunkable* is borrowed from [24,6]. These notions are closed under composition and identity.

Definition 3. *We define sub-categories of $\mathscr{D}$ as follows:*
$\mathscr{D}_l$ is the sub-category of linear morphisms of $\mathscr{D}$.
$\mathscr{D}_t$ ——————— thunkable morphisms of $\mathscr{D}$.
$\mathscr{N}_l$ ——————— linear morphisms of $\mathscr{N}$.
$\mathscr{P}_t$ ——————— thunkable morphisms of $\mathscr{P}$.

Observe that $\mathscr{N}$ and $\mathscr{N}_l$ are respectively the full sub-categories of $\mathscr{D}_t$ and $\mathscr{D}_l$ with negative objects. Symmetrically, $\mathscr{P}$ and $\mathscr{P}_t$ are respectively the full sub-categories of $\mathscr{D}_l$ and $\mathscr{D}_t$ whose objects are positive.

Proposition 4. *The hom-sets $\mathscr{D}(A,B)$ of a pre-duploid $\mathscr{D}$ extend to a (pro-)functor* $\boxed{\mathscr{D}(-,=) : {\mathscr{D}_t}^{\mathrm{op}} \times \mathscr{D}_l \to \boldsymbol{Set}}$ *defined for $f \in \mathscr{D}_t(A,B)$ and $g \in \mathscr{D}_l(C,D)$ with $\mathscr{D}(f,g) : \mathscr{D}(B,C) \to \mathscr{D}(A,D)$; $\mathscr{D}(f,g)(h) = g \circledcirc h \circledcirc f$.*

Proof. Restricting to f thunkable and g linear makes the definition unambiguous. Functoriality follows from $(g_1 \circledcirc g_2) \circledcirc h \circledcirc (f_2 \circledcirc f_1) = g_1 \circledcirc (g_2 \circledcirc h \circledcirc f_2) \circledcirc f_1$, which holds when f_1 and f_2 are thunkable and g_1 and g_2 are linear. ∎

2.2 Examples of Pre-duploids

Girard's Classical Logic. Girard's correlation spaces are a denotational semantics for classical logic. They do not form a category for lack of associativity of the composition [7,12]. However, they form a pre-duploid.

Blass Games. Blass [3] gives a game model for linear logic that fails to satisfy the associativity of composition. Thanks to Abramsky's analysis of this issue [1], we know that associativity fails due to composites of the form $N \to P \to M \to Q$. According to Abramsky, "*none of the other 15 polarisations give rise to a similar problem*". Therefore, Abramsky's formalisation of Blass games yields a pre-duploid. Thanks to Melliès's analysis of this so-called "*Blass problem*" [16], we know that the phenomenon is essentially the same as for Girard's classical logic.

Direct Models of Call by Value. Führmann [6] characterises the Kleisli category of a monad *via* the presence of a structure called *thunk*. In the contexts of models of call by value, the thunk implements laziness. Recall that a *thunk-force category* is a category $(\mathscr{P}, \bullet, \mathrm{id})$ together with a thunk $(L, \varepsilon, \vartheta)$ as defined next.

Definition 5 (Führmann). *A* thunk *on $\mathscr{P}$ is given by a functor $L : \mathscr{P} \to \mathscr{P}$ together with a natural transformation $\varepsilon : L \dot{\to} 1$ and a transformation $\vartheta : 1 \to L$ such that the transformation $\vartheta_L : L \to L^2$ is natural; satisfying the equations $\varepsilon \bullet \vartheta = \mathrm{id}$ and $L\varepsilon \bullet \vartheta_L = \mathrm{id}_L$ and $\vartheta_L \bullet \vartheta = L\vartheta \bullet \vartheta$.*

A thunk induces a comonad $(L, \varepsilon, \vartheta_L)$.

Observe that in a thunk-force category $(\mathscr{P}, \bullet, \mathrm{id}, L, \vartheta, \varepsilon)$, we can define a composite of $g : P \to Q$ and $f : LQ \to R$ with $g \circ f \overset{\text{def}}{=} g \bullet Lf \bullet \vartheta_P$. This compositions admits ε_P as a neutral element. This extends to a pre-duploid with compositions $\bullet$ and $\circ$ as follows. The positive objects are the objects of $\mathscr{P}$. The set of negative objects is given with $|\mathscr{N}| = {\Uparrow}|\mathscr{P}|$ for $\Uparrow$ a suitably chosen bijection with domain $|\mathscr{P}|$ (in other words $|\mathscr{N}|$ is a disjoint copy of $|\mathscr{P}|$). Then we take $\boxed{\mathscr{D}(A, B) \overset{\text{def}}{=} \mathscr{P}(\overline{A}, \underline{B})}$ where we define $\overline{P} = \underline{P} \overset{\text{def}}{=} P$ and $\overline{{\Uparrow}P} \overset{\text{def}}{=} LP$ and $\underline{{\Uparrow}P} \overset{\text{def}}{=} P$. With this definition, $\circ$ is a map $\mathscr{D}({\Uparrow}P, B) \times \mathscr{D}(A, {\Uparrow}P) \to \mathscr{D}(A, B)$ and ε_P is an element of $\mathscr{D}({\Uparrow}P, {\Uparrow}P)$. It is easy to check that this defines a pre-duploid.

In the context of λ_C models, this pre-duploid formalises how thunks implement laziness in call by value.

Now recall that Führmann calls thunkable any morphism $f \in \mathscr{P}(P, Q)$ such that $Lf \bullet \vartheta_P = \vartheta_Q \bullet f$. Not all morphisms of $\mathscr{P}$ are thunkable in general because ϑ is not necessarily natural. We can prove the following:

Proposition 6. *A morphism $f : P \to Q$ is thunkable in the sense of thunk-force categories if and only if it is thunkable in the sense of pre-duploids.*

Thus the transformation ϑ is natural if and only if the pre-duploid is a category (*i.e.* statisfies $\circ\bullet$-associativity).

Direct Models of Call by Name. The concept dual to Führmann's thunk is the one of *runnable monad*. A runnable monad on a category $\mathscr{C}$ is given by a functor $T : \mathscr{C} \to \mathscr{C}$ together with a natural transformation $\eta : 1 \dot{\to} T$ and a transformation $\rho : T \to 1$ such that the transformation $\rho_T : T^2 \to T$ is natural; satisfying the equations $\rho \circ \eta = \mathrm{id}$; $\rho_T \circ T\eta = \mathrm{id}_T$ and $\rho \circ T\rho = \rho \circ \rho_T$.

Runnable monads implement strictness in call by name. An example of a category with a runnable monad is given by Selinger's direct models of the call-by-name $\lambda\mu$ calculus [23]. Given a runnable monad, we can define, symmetrically to thunk-force categories above, a pre-duploid with a bijective map ${\Downarrow} : |\mathscr{N}| \to |\mathscr{P}|$.

2.3 Syntactic Pre-Duploid

The syntactic pre-duploid is given by a term syntax. It is made of terms (t) with a polarity which are identified up to β- and η-like equations. These equations are best described with auxiliary syntactic categories for evaluation contexts (e) and abstract machines ($c = \langle t \,\|\, e\rangle$); a technique that arose in the theory of control operators [4,8,2].

There are four sets of variables written $x^+, \alpha^+, x^\ominus, \alpha^\ominus$ to consider, and the following grammar ("…" indicates that we consider an extensible grammar):

$$\begin{aligned} t_+ &::= V_+ \mid \mu\alpha^+.c \mid \dots \\ t_\ominus &::= x^\ominus \mid \mu\alpha^\ominus.c \mid \dots \\ V_+ &::= x^+ \mid \dots \\ V &::= V_+ \mid t_\ominus \end{aligned}$$

(a) Terms and values

$$\begin{aligned} e_+ &::= \alpha^+ \mid \tilde\mu x^+.c \mid \dots \\ e_\ominus &::= \pi \mid \tilde\mu x^\ominus.c \mid \dots \\ \pi_\ominus &::= \alpha^\ominus \mid \dots \\ \pi &::= \pi_\ominus \mid e_+ \end{aligned}$$

(b) Contexts and stacks

$$c ::= \langle t_+ \,\|\, e_+\rangle \mid \langle t_\ominus \,\|\, e_\ominus\rangle$$

(c) Commands

Figure 1: The syntactic pre-duploid (*the variables that appear before a dot are bound*)

The binders are μ and $\tilde\mu$. They allow us to define composition as follows:

$$\boxed{\mathsf{let}\ x\ \mathsf{be}\ t\ \mathsf{in}\ u \overset{\text{def}}{=} \mu\alpha.\langle t \,\|\, \tilde\mu x.\langle u \,\|\, \alpha\rangle\rangle} \quad (\alpha \notin \mathrm{fv}(t,u))$$

In this macro-definition, the polarities of t and x must be the same, and the polarity of α and of "$\mathsf{let}\ x\ \mathsf{be}\ t\ \mathsf{in}\ u$" is determined by the one of u.

The contextual equivalence relation $\simeq$ determines the equality of morphisms. It is induced by the following rewrite rules:

$$\begin{aligned} \langle \mu\alpha.c \,\|\, \pi\rangle &\rhd c[\pi/\alpha] & t &\rhd \mu\alpha.\langle t \,\|\, \alpha\rangle\ (\alpha \notin \mathrm{fv}(t)) \\ \langle V \,\|\, \tilde\mu x.c\rangle &\rhd c[V/x] & e &\rhd \tilde\mu x.\langle x \,\|\, e\rangle\ (x \notin \mathrm{fv}(e)) \end{aligned}$$

The intuition is that positive terms are called by value while negative terms are called by name. Indeed we have:

$$\langle \mathsf{let}\ x\ \mathsf{be}\ V\ \mathsf{in}\ u \,\|\, \pi\rangle \rhd^* \langle u[V/x] \,\|\, \pi\rangle$$

but for a positive non-value t_+ instead of V, computation continues with t_+. This describes a call-by-value reduction. And with a negative non-stack $e_\ominus$ instead of π computation is delayed until a stack (a linear evaluation context) is reached. This describes a call-by-name reduction. In the latter case, "$\mathsf{let}\ x\ \mathsf{be}\ V\ \mathsf{in}\ u$" and therefore u are negative.

Among other equations, we have for all terms and variables:

$$\mathsf{let}\ x\ \mathsf{be}\ y\ \mathsf{in}\ t \simeq t[y/x] \tag{1}$$

$$\mathsf{let}\ x\ \mathsf{be}\ t\ \mathsf{in}\ x \simeq t \tag{2}$$

$$\mathsf{let}\ y^+\ \mathsf{be}\ (\mathsf{let}\ x\ \mathsf{be}\ t\ \mathsf{in}\ u_+)\ \mathsf{in}\ v \simeq \mathsf{let}\ x\ \mathsf{be}\ t\ \mathsf{in}\ \mathsf{let}\ y^+\ \mathsf{be}\ u_+\ \mathsf{in}\ v \tag{3}$$

$$\mathsf{let}\ y\ \mathsf{be}\ (\mathsf{let}\ x^\ominus\ \mathsf{be}\ t_\ominus\ \mathsf{in}\ u)\ \mathsf{in}\ v \simeq \mathsf{let}\ x^\ominus\ \mathsf{be}\ t_\ominus\ \mathsf{in}\ \mathsf{let}\ y\ \mathsf{be}\ u\ \mathsf{in}\ v \tag{4}$$

The syntactic pre-duploid gives rise to a pre-duploid with two objects $+$ and $\ominus$ and with formal objects $(x \mapsto t) : \varepsilon_1 \to \varepsilon_2$ as morphisms, where ε_1 and ε_2 determine the polarities of x and t (respectively). Equations (1) and (2) mean that modulo α-conversion, variables provide a neutral element for the composition. Equations (3) and (4) correspond respectively to $\bullet\odot$- and $\odot\circ$- associativity.

It is not possible to rewrite "$\mathsf{let}\ y^{\ominus}\ \mathsf{be}\ (\mathsf{let}\ x^{+}\ \mathsf{be}\ t_{+}\ \mathsf{in}\ u_{\ominus})\ \mathsf{in}\ v$" into "$\mathsf{let}\ x^{+}\ \mathsf{be}\ t_{+}\ \mathsf{in}\ \mathsf{let}\ y^{\ominus}\ \mathsf{be}\ u_{\ominus}\ \mathsf{in}\ v$". In other words, without imposing additional equations, we have in general $h \circ (g \bullet f) \neq (h \circ g) \bullet f$.

3 Duploids

We now enrich pre-duploids with operators of polarity coercion $\Downarrow$, $\Uparrow$ called *shifts*.[2]

Definition 7. *A* duploid *is a pre-duploid $\mathscr{D}$ given with mappings $\Downarrow : |\mathscr{N}| \to |\mathscr{P}|$ and $\Uparrow : |\mathscr{P}| \to |\mathscr{N}|$, together with, for all $P \in |\mathscr{P}|$ and $N \in |\mathscr{N}|$, morphisms subject to equations:*

$$\boxed{\begin{array}{rl} \mathsf{delay}_P & : P \to \Uparrow P \\ \mathsf{force}_P & : \Uparrow P \to P \\ \mathsf{wrap}_N & : N \to \Downarrow N \\ \mathsf{unwrap}_N & : \Downarrow N \to N \end{array}} \quad \boxed{\begin{array}{rl} \mathsf{force}_P \circ (\mathsf{delay}_P \bullet f) & = f \ (\forall f \in \mathscr{D}(A, P)) \\ (f \circ \mathsf{unwrap}_N) \bullet \mathsf{wrap}_N & = f \ (\forall f \in \mathscr{D}(N, A)) \\ \mathsf{delay}_P \bullet \mathsf{force}_P & = \mathrm{id}_{\Uparrow P} \\ \mathsf{wrap}_N \circ \mathsf{unwrap}_N & = \mathrm{id}_{\Downarrow N} \end{array}}$$

Proposition 8. *For any N, wrap_N is thunkable. Dually, for any P, force_P is linear.*

Proof. For all g, h we have $h \circ (g \bullet \mathsf{wrap}_N) = (h \circ (g \bullet \mathsf{wrap}_N) \circ \mathsf{unwrap}_N) \bullet \mathsf{wrap}_N = (h \circ (g \bullet \mathsf{wrap}_N \circ \mathsf{unwrap}_N)) \bullet \mathsf{wrap}_N = (h \circ g) \bullet \mathsf{wrap}_N$. Hence wrap_N is linear. The other result follows by symmetry. ∎

Thus we have the following equivalent definition of a duploid:

Definition 9. *A* duploid *is a pre-duploid $\mathscr{D}$ given with mappings $\Downarrow : |\mathscr{N}| \to |\mathscr{P}|$ and $\Uparrow : |\mathscr{P}| \to |\mathscr{N}|$, together with a family of invertible linear maps $\mathsf{force}_P : \Uparrow P \to P$ and a family of invertible thunkable maps $\mathsf{wrap}_N : N \to \Downarrow N$.*

3.1 Syntactic Duploid

Let us start with the syntax, with which we provide computational intuitions for the shifts. The syntactic duploid extends the syntactic pre-duploid with a type $\Uparrow P$ of suspended strict computations, and a type $\Downarrow N$ of lazy computations encapsulated into a value. Then $\mathsf{delay} \bullet f$ represents the suspended strict computation f and the inverse operation force triggers the evaluation of its argument (this is why it is linear in its negative argument). The morphism $\mathsf{wrap} \circ f$ represents f encapsulated into a value (this is why it is thunkable) and unwrap removes the encapsulation.

We extend the syntactic pre-duploid as follows:

[2] Our notation is reminiscent of Melliès [16].

$$V_+ ::= \ldots \mid \{t_\ominus\} \mid \ldots \qquad\qquad e_+ ::= \ldots \mid \tilde{\mu}\{x^\ominus\}.c \mid \ldots$$

$$t_\ominus ::= \ldots \mid \mu\{\alpha^+\}.c \mid \ldots \qquad\qquad \pi_\ominus ::= \ldots \mid \{e_+\} \mid \ldots$$

(a) Terms and values | (b) Contexts and stacks

Figure 2: The syntactic duploid (extending the syntactic pre-duploid)

We also extend the relation $\rhd$ with the following rules:

$$\boxed{\begin{array}{ll} \langle\{t_\ominus\} \,\|\, \tilde{\mu}\{x^\ominus\}.c\rangle \rhd c[t_\ominus/x^\ominus] & e_+ \rhd \tilde{\mu}\{x^\ominus\}.\langle\{x^\ominus\} \,\|\, e_+\rangle \ (x^\ominus \notin \mathrm{fv}(e_+)) \\ \langle\mu\{\alpha^+\}.c \,\|\, \{e_+\}\rangle \rhd c[e_+/\alpha^+] & t_\ominus \rhd \mu\{\alpha^+\}.\langle t_\ominus \,\|\, \{\alpha^+\}\rangle \ (\alpha^+ \notin \mathrm{fv}(t_\ominus)) \end{array}}\,.$$

The new constructions add to the syntax of terms the following operations (in addition to values $\{t_\ominus\}$):

$$\begin{aligned} \mathsf{let}\ \{x^\ominus\}\ \mathsf{be}\ t_+\ \mathsf{in}\ u &\stackrel{\text{def}}{=} \mu\alpha.\langle t_+ \,\|\, \tilde{\mu}\{x^\ominus\}.\langle u \,\|\, \alpha\rangle\rangle \\ \mathsf{delay}(t_+) &\stackrel{\text{def}}{=} \mu\{\alpha^+\}.\langle t_+ \,\|\, \alpha^+\rangle \\ \mathsf{force}(t_\ominus) &\stackrel{\text{def}}{=} \mu\alpha^+.\langle t_\ominus \,\|\, \{\alpha^+\}\rangle \end{aligned}$$

We have in particular:

$$\begin{array}{cc} \mathsf{let}\ \{x^\ominus\}\ \mathsf{be}\ \{t_\ominus\}\ \mathsf{in}\ u \simeq u[t_\ominus/x^\ominus] & \mathsf{let}\ \{x^\ominus\}\ \mathsf{be}\ t_+\ \mathsf{in}\ \{x^\ominus\} \simeq t_+ \\ \mathsf{force}(\mathsf{delay}(t_+)) \simeq t_+ & \mathsf{delay}(\mathsf{force}(t_\ominus)) \simeq t_\ominus\,. \end{array}$$

We can show that this extends the syntactic pre-duploid into a duploid (with wrap and unwrap interpreted as $x^\ominus \mapsto \{x^\ominus\}$ and $x^+ \mapsto \mathsf{let}\ \{y^\ominus\}\ \mathsf{be}\ x^+\ \mathsf{in}\ y^\ominus$, respectively).

3.2 The Duploid Construction

Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be two categories and $F \dashv G : \mathscr{C}_1 \to \mathscr{C}_2$ an adjunction given by natural transformations $\sharp : \mathscr{C}_1(F-,=) \to \mathscr{C}_2(-,G=)$ and $\flat = \sharp^{-1}$. Note $G : \mathscr{C}_1 \to \mathscr{C}_2$.

The goal of the duploid construction is to define a notion of morphisms $A \to B$ for A and B objects of *either* category $\mathscr{C}_1$ and $\mathscr{C}_2$. Let us introduce the convention that objects of $\mathscr{C}_1$ are negative and written N, M..., while the objects of $\mathscr{C}_2$ are positive and written P, Q... Also, we write $\bullet$ the composition in $\mathscr{C}_1$ and $\circ$ the composition in $\mathscr{C}_2$.

We first define *oblique morphisms* $P \to_{\mathscr{D}} N$, with $P \in |\mathscr{C}_2|$ and $N \in |\mathscr{C}_1|$, equivalently as maps $P \to GN$ or $FP \to N$ (thanks to the isomorphism $\sharp$). Then we observe that oblique morphisms compose either in $\mathscr{C}_1$ or in $\mathscr{C}_2$ as follows:

$$\dfrac{\dfrac{f : P \to_{\mathscr{D}} FQ}{f : FP \to FQ} \qquad \dfrac{g : Q \to_{\mathscr{D}} N}{g : FQ \to N}}{\dfrac{g \bullet f : FP \to N}{g \bullet f : P \to_{\mathscr{D}} N}} \qquad \dfrac{\dfrac{f : P \to_{\mathscr{D}} N}{f : P \to GN} \qquad \dfrac{g : GN \to_{\mathscr{D}} M}{g : GN \to GM}}{\dfrac{g \circ f : P \to GM}{g \circ f : P \to_{\mathscr{D}} M}}$$

Thus we define morphisms $A \to_{\mathscr{D}} B$ as oblique morphisms:

$$\boxed{A^+ \to_{\mathscr{D}} B^\ominus}, \qquad \text{where} \qquad \begin{array}{ll} P^+ \stackrel{\text{def}}{=} P & P^\ominus \stackrel{\text{def}}{=} FP \\ N^+ \stackrel{\text{def}}{=} GN & N^\ominus \stackrel{\text{def}}{=} N \end{array}$$

In other words, we define $|\mathscr{D}| \overset{\text{def}}{=} |\mathscr{C}_1| \uplus |\mathscr{C}_2|$ and (taking an irrelevant bias towards $\mathscr{C}_1$) we define $\mathscr{D}(A,B) \overset{\text{def}}{=} \mathscr{C}_1(FA^+, B^\ominus)$. Positive composition is given by the composition in $\mathscr{C}_1$. Composition of $f \in \mathscr{D}(A,N)$ and $g \in \mathscr{D}(N,B)$ is given by $g \circ^{\mathscr{D}} f \overset{\text{def}}{=} (g^\sharp \circ^{\mathscr{C}_2} f^\sharp)^\flat$. Identities are given with $\mathrm{id}_P^{\mathscr{D}} \overset{\text{def}}{=} \mathrm{id}_{FP}^{\mathscr{C}_1}$ and $\mathrm{id}_N^{\mathscr{D}} \overset{\text{def}}{=} {\mathrm{id}_{GN}^{\mathscr{C}_2}}^\flat$.

Proposition 10. *The above defines a pre-duploid $\mathscr{D}$.*

Proof (sketch). $\bullet\bullet$-associativity is given, and $\circ\circ$-associativity is immediate using the fact that $\sharp$ and $\flat$ are inverse. $\bullet\circ$-associativity relies on the fact that the transformations $\sharp$ and $\flat$ are natural. ∎

Remark 11. In particular $\mathscr{P}$ is the Kleisli category $(\mathscr{C}_2)_{GF}$ of the monad GF and $\mathscr{N}$ is the Kleisli category $(\mathscr{C}_1)_{FG}$ of the co-monad FG.

The pre-duploid has shifts, defined as follows:

$$\Uparrow P \overset{\text{def}}{=} FP \qquad \Downarrow N \overset{\text{def}}{=} GN$$
$$\mathscr{D}(P, \Uparrow P) \ni \mathsf{delay}_P \overset{\text{def}}{=} \mathrm{id}_{FP}^{\mathscr{C}_1} \in \mathscr{C}_1(FP, FP)$$
$$\mathscr{D}(\Uparrow P, P) \ni \mathsf{force}_P \overset{\text{def}}{=} (\mathrm{id}_{GFP})^\flat \in \mathscr{C}_1(FGFP, FP)$$
$$\mathscr{D}(N, \Downarrow N) \ni \mathsf{wrap}_N \overset{\text{def}}{=} \mathrm{id}_{FGN}^{\mathscr{C}_1} \in \mathscr{C}_1(FGN, FGN)$$
$$\mathscr{D}(\Downarrow N, N) \ni \mathsf{unwrap}_N \overset{\text{def}}{=} (\mathrm{id}_{GN})^\flat \in \mathscr{C}_1(FGN, N)$$

It is easy to see that:

Proposition 12. *Every adjunction determines a duploid as above.*

3.3 Linear and Thunkable Morphisms in Duploids

In duploids, we have the following useful characterisation of linear and thunkable morphisms.

Proposition 13. *In a duploid $\mathscr{D}$, let $f \in \mathscr{D}(A,P)$. Then f is thunkable if and only if:*

$$(\mathsf{wrap}_{\Uparrow P} \circ \mathsf{delay}_P) \bullet f = \mathsf{wrap}_{\Uparrow P} \circ (\mathsf{delay}_P \bullet f) \tag{5}$$

Dually, let $f \in \mathscr{D}(N,B)$. Then f is linear if and only if:

$$f \circ (\mathsf{unwrap}_N \bullet \mathsf{force}_{\Downarrow N}) = (f \circ \mathsf{unwrap}_N) \bullet \mathsf{force}_{\Downarrow N}$$

Proof. We establish the non-trivial implication for the first case. The second case is obtained by symmetry. First we prove that any morphism that satisfies (5) also satisfies $(h \circ \mathsf{delay}_P) \bullet f = h \circ (\mathsf{delay}_P \bullet f)$ for any $h \in \mathscr{D}(\Uparrow P, A)$. Indeed for any such h we have:

$$\begin{aligned}(h \circ \mathsf{delay}_P) \bullet f &= (h \circ \mathsf{unwrap}_{\Uparrow P}) \bullet \underline{(\mathsf{wrap}_{\Uparrow P} \circ \mathsf{delay}_P) \bullet f} \\ &= (h \circ \underline{\mathsf{unwrap}_{\Uparrow P}) \bullet \mathsf{wrap}_{\Uparrow P}} \circ (\mathsf{delay}_P \bullet f) \quad \text{(by hypothesis)} \\ &= h \circ (\mathsf{delay}_P \bullet f)\end{aligned}$$

Now we prove that f is thunkable. For any g, h we have:

$$\begin{aligned}
(h \circ \underline{g}) \bullet f &= \underline{(h \circ (g \bullet \mathsf{force}_P) \circ \mathsf{delay}_P) \bullet f} \\
&= h \circ \underline{(g \bullet \mathsf{force}_P) \circ (\mathsf{delay}_P \bullet f)} && \text{(with the above)} \\
&= h \circ ((g \bullet \underline{\mathsf{force}_P \circ \mathsf{delay}_P}) \bullet f) && \text{(with the above again)} \\
&= h \circ (g \bullet f) && \blacksquare
\end{aligned}$$

By applying the above proposition to the duploid construction, we easily deduce the following:

Proposition 14. *Let $F \dashv_{(\eta,\varepsilon)} G : \mathscr{C}_1 \to \mathscr{C}_2$ be an adjunction, and consider the associated duploid $\mathscr{D}$. Then $f \in \mathscr{D}(N, A)$ is linear if and only if $f \circ \varepsilon_{FGN} = f \circ FG\varepsilon_N$ (in $\mathscr{C}_1$), and $f \in \mathscr{D}(A, P)$ is thunkable if and only if its transpose $f^\sharp \in \mathscr{C}_2(A^+, GFP)$ satisfies $\eta_{GFP} \circ f^\sharp = GF\eta_P \circ f^\sharp$ (in $\mathscr{C}_2$).*

Now recall that an adjunction $F \dashv G$ that satisfies either of the following equivalent statements is called *idempotent*: the multiplication of the associated monad is an isomorphism; or the co-multiplication of the associated co-monad is an isomorphism; or we have $\varepsilon_{GF} = GF\varepsilon$; or we have $\eta_{FG} = FG\eta$. Thus we deduce the following:

Corollary 15. *Let $F \dashv_{(\eta,\varepsilon)} G : \mathscr{C}_1 \to \mathscr{C}_2$. The associated duploid $\mathscr{D}$ is a category if and only if the adjunction is idempotent.*

3.4 Structure of Shifts

As we have seen, the Kleisli category of a co-monad is described by a runnable monad; and the Klesli category of a monad is described by a thunk, which is a co-monad. We observe a similar phenomenon with duploids. We show that there is a reversed adjunction, in the sense that the right adjoint $\Uparrow$ is from positives to negatives:

$$\boxed{\Downarrow \dashv \Uparrow : \mathscr{P} \to \mathscr{N}}$$

Actually, we state a wider adjunction. First remark that we can extend the shifts $\Downarrow, \Uparrow$ to all objects in a straightforward manner:

$$\Downarrow A \overset{\text{def}}{=} \begin{cases} \Downarrow N & \text{if } A = N \\ P & \text{if } A = P \end{cases} \qquad \begin{aligned} \mathsf{delay}_N &\overset{\text{def}}{=} \mathrm{id}_N : N \to \Uparrow N \\ \mathsf{force}_N &\overset{\text{def}}{=} \mathrm{id}_N : \Uparrow N \to N \end{aligned}$$

$$\Uparrow A \overset{\text{def}}{=} \begin{cases} N & \text{if } A = N \\ \Uparrow P & \text{if } A = P \end{cases} \qquad \begin{aligned} \mathsf{wrap}_P &\overset{\text{def}}{=} \mathrm{id}_P : P \to \Downarrow P \\ \mathsf{unwrap}_P &\overset{\text{def}}{=} \mathrm{id}_P : \Downarrow P \to P \end{aligned}$$

By "extend", we mean that we have for all f, g:

$$\begin{aligned} (f \odot \mathsf{force}_A) \circ (\mathsf{delay}_A \odot g) &= f \odot g & \qquad \mathsf{delay}_A \odot \mathsf{force}_A &= \mathrm{id}_A \\ (f \odot \mathsf{unwrap}_A) \bullet (\mathsf{wrap}_A \odot g) &= f \odot g & \qquad \mathsf{wrap}_A \odot \mathsf{unwrap}_A &= \mathrm{id}_A \end{aligned}$$

Also, extending proposition 8, we have, for all objects A, that unwrap_A and wrap_A are thunkable whereas delay_A and force_A are linear.

Proposition 16. *Let $\mathscr{D}$ be a duploid. The following:*

$$\Uparrow f \overset{\text{def}}{=} \mathsf{delay}_B \odot f \odot \mathsf{force}_A \qquad \Downarrow f \overset{\text{def}}{=} \mathsf{wrap}_B \odot f \odot \mathsf{unwrap}_A$$

define functors $\Uparrow : \mathscr{D}_l \to \mathscr{N}_l$ and $\Downarrow : \mathscr{D}_t \to \mathscr{P}_t$ that take part in adjoint equivalences of categories $I \dashv_{(\mathsf{delay},\mathsf{force})} \Uparrow : \mathscr{D}_l \to \mathscr{N}_l$ and $I \dashv_{(\mathsf{wrap},\mathsf{unwrap})} \Downarrow : \mathscr{D}_t \to \mathscr{P}_t$, where I denotes the inclusion functors.

Proof (sketch). The result follows from the fact that delay and force are inverse natural transformations in $\mathscr{D}_l$; likewise for wrap and unwrap in $\mathscr{D}_t$. ■

We can deduce the following:

Proposition 17. *Let $\mathscr{D}$ be a duploid. We have natural isomorphisms between (pro-)functors $\boxed{\mathscr{D}_t(-, I\Uparrow =) \simeq \mathscr{D}(-,=) \simeq \mathscr{D}_l(I\Downarrow -, =) : {\mathscr{D}_t}^{\mathrm{op}} \times \mathscr{D}_l \to \mathbf{Set}}$ where I denotes the inferrable inclusion functors.*

In particular, leaving the inclusion functors implicit, we have the adjunctions:

$$\begin{tikzcd} \mathscr{D}_t \arrow[r, bend left, "\Downarrow"] \arrow[r, phantom, "\perp"] & \mathscr{D}_l \arrow[l, bend left, "\Uparrow"] \end{tikzcd} \qquad \begin{tikzcd} \mathscr{N} \arrow[r, bend left, "\Downarrow"] \arrow[r, phantom, "\perp"] & \mathscr{P} \arrow[l, bend left, "\Uparrow"] \end{tikzcd}$$

The adjunction $\Downarrow \dashv \Uparrow$ distinguishes our interpretation of polarities from ones based on adjunctions of the form $\uparrow \dashv \downarrow$ that appears in the context of focusing in logic and continuation-passing style in programming (see Laurent [11], Zeilberger [25]). Our direct notion of polarities adds a level of granularity. In terms of continuations, our polarities makes the distinction between continuations that are meant to be applied and continuations that are meant to be passed.

3.5 The Category of Duploids

Definition 18. *A* functor of pre-duploids *$F : \mathscr{D}_1 \to \mathscr{D}_2$ is given by a mapping on objects $|F| : |\mathscr{D}_1| \to |\mathscr{D}_2|$ that preserves polarities, together with mappings on morphisms $F_{A,B} : \mathscr{D}_1(A,B) \to \mathscr{D}_2(FA,FB)$, satisfying $F\mathrm{id}_A = \mathrm{id}_{FA}$ and $F(g \odot f) = Fg \odot Ff$. A* functor of duploids *$F : \mathscr{D}_1 \to \mathscr{D}_2$ is a functor of pre-duploids such that $F\mathsf{force}_P$ is linear for all $P \in |\mathscr{P}_1|$, and $F\mathsf{wrap}_N$ is thunkable for all $N \in |\mathscr{N}_1|$.*

Proposition 19. *Let $\mathscr{D}$ and $\mathscr{D}'$ be two duploids and let $F : \mathscr{D} \to \mathscr{D}'$ be a mapping on objects $|F| : |\mathscr{D}| \to |\mathscr{D}'|$ that preserves polarities, together with mappings on morphisms $F_{A,B} : \mathscr{D}(A,B) \to \mathscr{D}'(FA,FB)$. Then F is a functor of duploids if and only if F restricts to functors $F_t : \mathscr{D}_t \to \mathscr{D}'_t$ and $F_l : \mathscr{D}_l \to \mathscr{D}'_l$, such that the transformation $F : \mathscr{D}(-,=) \to \mathscr{D}'(F_t -, F_l =)$ is natural.*

Proof. ($\Leftarrow$) is easy to prove. ($\Rightarrow$): Suppose that F is a functor of duploids. First we establish that the full sub-pre-duploid $F\mathscr{D}$ of $\mathscr{D}'$ with objects of the form FA for $A \in |\mathscr{D}|$ has a duploid structure given by Fdelay, Fforce, Fwrap and Funwrap. This follows from definition 9, using the hypothesis that Fforce is linear and Fwrap is thunkable.

Then, considering proposition 13 applied to the duploid $F\mathscr{D}$, we show that F preserves linearity and thunkability. In other words it restricts to functors $F_t : \mathscr{D}_t \to \mathscr{D}'_t$ and $F_l : \mathscr{D}_l \to \mathscr{D}'_l$. That $F : \mathscr{D}(-,=) \to \mathscr{D}'(F_t-, F_l=)$ is a natural transformation follows from $Fh \circledcirc Fg \circledcirc Ff = F(h \circledcirc g \circledcirc f)$ which makes sense for h linear and f thunkable. ■

Definition 20. *$\mathfrak{Dupl}$ is the category whose objects are duploids and whose morphisms are duploid functors. The obvious identity in $\mathfrak{Dupl}$ is written $1_{\mathscr{D}}$.*

3.6 Examples of Duploids

The Blass Phenomenon in Conway Games. Melliès [16] comes close to building a duploid using the construction of Blass games. According to his analysis [16, Section 3], the Blass problem comes down to the fact that the (pro-)functor $\mathscr{C}_1(F-,=) : \mathscr{C}_2{}^{\mathrm{op}} \times \mathscr{C}_1 \to \boldsymbol{Set}$, in the terminology of Section 3.2, does not extend into a functor $\mathscr{P}^{\mathrm{op}} \times \mathscr{N} \to \boldsymbol{Set}$ where $\mathscr{P}$ and $\mathscr{N}$ are respectively the Kleisli categories of the monad GF and the co-monad FG. This is the essence of proposition 15. He then defines a category for an asynchronous variant of Conway games. As he shows, asynchronism is a way to force the double-negation monad to be idempotent, and therefore to recover associativity of composition. He builds this way a game model of linear logic.

Girard's Polarisation. Girard's polarised translation of the classical logic **LC** into intuitionistic logic [7], further formulated by Danos, Joinet and Schellinx [5] and Laurent [11], inspired the duploid construction. Girard's translation corresponds to considering in the duploid construction the self-adjunction of the negation functor $\neg = R^-$ in $\boldsymbol{Set}$ for R arbitrary. But obviously, the duploid obtained from the self-adjunction of negation in any response category (in the terminology of Selinger [23]) gives a denotational semantics of **LC**. Thielecke [24] later noticed the importance of this self-adjunction in the understanding of continuation-passing style.

Response categories have recently been refined into dialogue categories by Melliès and Tabareau [17] to provide a denotational semantics of linear logic *via* the polarised translation. This was conceived as an abstract account of the asynchronous games of Melliès [16] mentioned above.

Direct Models of Call by Value and of Call by Name. We defined a pre-duploid with a bijection $\Uparrow : |\mathscr{P}| \to |\mathscr{N}|$ from a thunk-force category $(\mathscr{P}, \bullet, \mathrm{id}, L, \vartheta, \varepsilon)$. We complete the definition into a duploid by defining $\Downarrow : |\mathscr{N}| \to |\mathscr{P}|$ with $\Downarrow\Uparrow P \overset{\mathrm{def}}{=} LP$; and delay, force, wrap, unwrap in an obvious manner. We can show that thunk-force categories are characterised as duploids where $\Uparrow$ is bijective on objects. Symmetrically, we can show that categories with a runnable monad are characterised as duploids where $\Downarrow$ is bijective on objects.

4 Structure Theorem

4.1 Every Duploid Comes From an Adjunction

Proposition 21. *Let $\mathscr{D}$ be a duploid. We define $\uparrow : \mathscr{P}_t \to \mathscr{N}_l$ the restriction of $\Uparrow$ and $\downarrow : \mathscr{N}_l \to \mathscr{P}_t$ the restriction of $\Downarrow$. There is an adjunction $\uparrow \dashv \downarrow$ with unit* $\mathsf{wrap}_{\Uparrow} \circ \mathsf{delay}$ *and co-unit* $\mathsf{unwrap} \bullet \mathsf{force}_{\Downarrow}$.

Proof. Due to the adjoint equivalences from proposition 16, we have the following natural isomorphisms:

$$\mathscr{N}_l(\Uparrow I-,=) \simeq \mathscr{D}_l(I-,I=) : \mathscr{P}_t{}^{\mathrm{op}} \times \mathscr{N}_l \to \boldsymbol{Set}$$
$$\mathscr{P}_t(-,\Downarrow I=) \simeq \mathscr{D}_t(I-,I=) : \mathscr{P}_t{}^{\mathrm{op}} \times \mathscr{N}_l \to \boldsymbol{Set}\,,$$

where I denotes the inferrable inclusion functors. Since we have $\mathscr{D}_l(P,N) = \mathscr{D}(P,N) = \mathscr{D}_t(P,N)$, we also have $\mathscr{D}_l(I-,I=) = \mathscr{D}_t(I-,I=)$ above. Thus we have a natural isomorphism $\mathscr{N}_l(\Uparrow-,=) = \mathscr{N}_l(\Uparrow I-,=) \simeq \mathscr{P}_t(-,\Downarrow I=) = \mathscr{P}_t(-,\downarrow=)$. We can check that the unit is $\mathsf{wrap}_{\Uparrow} \circ \mathsf{delay}$ and the co-unit is $\mathsf{unwrap} \bullet \mathsf{force}_{\Downarrow}$. ■

Proposition 22. *There is an isomorphism between $\mathscr{D}$ and the duploid $\mathscr{D}'$ obtained from the above adjunction $\Uparrow \dashv \downarrow$.*

Proof (sketch). Recall that $\mathscr{D}'$ is defined with $|\mathscr{D}'| = |\mathscr{D}|$ and $\mathscr{D}'(A,B) = \mathscr{N}_l(\Uparrow\Downarrow A, \Uparrow B)$. According to propositions 16 and 17, we have natural isomorphisms $\mathscr{D}(-,=) \simeq \mathscr{D}_l(I\Downarrow-,=) \simeq \mathscr{N}_l(\Uparrow\Downarrow-,\Uparrow=)$, and thus for all $A,B \in |\mathscr{D}|$ we have a bijection $\mathscr{D}(A,B) \to \mathscr{D}'(A,B)$. It is easy to verify that this mapping defines a functor of duploids $F : \mathscr{D} \to \mathscr{D}'$. Using the characterisation of proposition 19, its inverse is a functor of duploids. ■

4.2 The Equalising Requirement

Definition 23. *An adjunction $F \dashv_{(\eta,\varepsilon)} G : \mathscr{C}_1 \to \mathscr{C}_2$ satisfies the* equalising requirement *if and only if for all $P \in |\mathscr{C}_2|$, η_P is an equaliser of η_{GFP} and $GF\eta_P$, and for all $N \in |\mathscr{C}_1|$, ε_N is a co-equaliser of ε_{FGN} and $FG\varepsilon_N$.*

We give an equivalent formulation of this condition in terms of the associated duploid:

Proposition 24. *Let $F \dashv_{(\eta,\varepsilon)} G : \mathscr{C}_1 \to \mathscr{C}_2$ be an adjunction, and consider the associated duploid $\mathscr{D}$. The adjunction satisfies the equalising requirement if and only if for all objects A,P,N the following three conditions hold:*

1. *ε_N is an epimorphism and η_P is a monomorphism; or equivalently G and F are faithful;*
2. *all linear morphisms $f \in \mathscr{D}(N,A)$ are of the form $g \circ \varepsilon_N$ with $g \in \mathscr{C}_1(N,A^-)$; or equivalently all linear morphisms are in the image of G modulo the adjunction;*
3. *all thunkable morphisms $f \in \mathscr{D}(A,P)$ are (modulo the adjunction) of the form $\eta_P \circ g$ with $g \in \mathscr{C}_2(A^+,P)$; or equivalently all thunkable morphisms are in the image of F;*

Proof (sketch). Follows from the characterisation in proposition 14. ■

Proposition 25. *Let $\mathscr{D}$ be a duploid and consider the adjunction $\Uparrow \dashv \downarrow : \mathscr{N}_l \to \mathscr{P}_t$. The adjunction satisfies the equalising requirement.*

Proof (sketch). Follows easily from the fact that $\varepsilon = \mathsf{unwrap} \bullet \mathsf{force}_{\Downarrow}$ has a section in $\mathscr{N}$, namely $\mathsf{delay}_{\Downarrow} \bullet \mathsf{wrap} : 1 \stackrel{\cdot}{\to} \Uparrow\Downarrow$, and symmetrically for η. ■

4.3 Main Result

We consider pseudo maps of adjunctions as defined by Jacobs [9]:

Definition 26. *Let $F \dashv_{(\eta,\varepsilon)} G : \mathscr{C}_1 \to \mathscr{C}_2$ and $F' \dashv_{(\eta',\varepsilon')} G' : \mathscr{C}'_1 \to \mathscr{C}'_2$ be two adjunctions. A* pseudo map of adjunctions*:*

$$\boxed{(H_1, H_2, \phi, \psi) : (F \dashv_{(\eta,\varepsilon)} G) \to (F' \dashv_{(\eta',\varepsilon')} G')}$$

is given by a pair of functors $H_1 : \mathscr{C}_1 \to \mathscr{C}'_1$ and $H_2 : \mathscr{C}_2 \to \mathscr{C}'_2$ together with natural isomorphisms $\phi : F'H_2 \overset{\simeq}{\to} H_1F$ and $\psi : G'H_1 \overset{\simeq}{\to} H_2G$, such that H_1 and H_2 preserve η and ε up to isomorphism: $H_2\eta = \psi_F \circ G'\phi \circ \eta'_{H_2}$ and $H_1\varepsilon = \varepsilon'_{H_1} \circ F'\psi^{-1} \circ \phi_G^{-1}$.

As noted by Jacobs, two pseudo maps (H_1, H_2, ϕ, ψ) and $(H'_1, H'_2, \phi', \psi')$ compose as:

$$(H'_1, H'_2, \phi', \psi') \circ (H_1, H_2, \phi, \psi) = (H'_1H_1, H'_2H_2, H'_1\phi \circ \phi'_{H_2}, H'_2\psi \circ \psi'_{H_1})\,.$$

Definition 27. *The category of adjunctions $\mathcal{A}\boldsymbol{dj}$ has adjunctions between locally small categories as objects and pseudo maps of adjunctions as morphisms. The full subcategory $\mathcal{A}\boldsymbol{dj}_{\mathbf{eq}}$ of $\mathcal{A}\boldsymbol{dj}$ consists in adjunctions that satisfy the equalising requirement.*

Theorem 28. *There are a reflection and an equivalence as follows:*

$$\mathcal{D}\boldsymbol{upl} \simeq \mathcal{A}\boldsymbol{dj}_{\mathbf{eq}} \triangleleft \mathcal{A}\boldsymbol{dj}$$

Proof (sketch). The functor $j : \mathcal{A}\boldsymbol{dj} \to \mathcal{D}\boldsymbol{upl}$ is given on objects by the duploid construction. The functor $i : \mathcal{D}\boldsymbol{upl} \to \mathcal{A}\boldsymbol{dj}_{\mathbf{eq}}$ is given on objects by proposition 25. Proposition 22 gives the family of isomorphisms $ji\mathscr{D} \simeq \mathscr{D}$.

The complete proof appears in the author's PhD thesis, Chapter II [20].

Intuitively, theorem 28 together with proposition 24 mean that the duploid construction j completes the values with all the expressions that are pure, and completes the stacks with all the evaluation contexts that are linear. Moreover j identifies any two values that denote the same expression, and any two stacks that denote the same evaluation context.

5 Ongoing Work

This work was developed during a collaboration with Marcelo Fiore and Pierre-Louis Curien, in an effort to connect the **L** system [4,8,19,20] with adjunction models. The calculus suggests that connectives should have an elegant characterisation in terms of duploids, which is the subject of an ongoing work.

Acknowledgements. I am grateful to Pierre-Louis Curien and Marcelo Fiore for their influence and support, as well as for their careful reading and suggestions. I am grateful to Paul-André Melliès for the many discussions. For discussions relevant to this article I also wish to thank Jean-Yves Girard, Hugo Herbelin and Olivier Danvy. Thanks to the anonymous reviewers for their suggestions. This work has been partially supported by the French National Research Agency[3] and by the Fondation Sciences Mathématiques de Paris.

[3] Projects Choco (ANR-07-BLAN-0324), Coquas (ANR-12-JS02-006-01), Logoi (ANR-10-BLAN-0213) and Recre (ANR-11-BS02-0010).

References

1. Abramsky, S.: Sequentiality vs. concurrency in games and logic. Math. Struct. Comput. Sci. 13(4), 531–565 (2003)
2. Ariola, Z.M., Herbelin, H.: Control Reduction Theories: the Benefit of Structural Substitution. Journal of Functional Programming 18(3), 373–419 (May 2008)
3. Blass, A.: A game semantics for linear logic. Ann. Pure Appl. Logic 56(1-3), 183–220 (1992)
4. Curien, P.L., Herbelin, H.: The duality of computation. ACM SIGPLAN Notices 35, 233–243 (2000)
5. Danos, V., Joinet, J.B., Schellinx, H.: A New Deconstructive Logic: Linear Logic. Journal of Symbolic Logic 62 (3), 755–807 (1997)
6. Führmann, C.: Direct Models for the Computational Lambda Calculus. Electr. Notes Theor. Comput. Sci. 20, 245–292 (1999)
7. Girard, J.Y.: A new constructive logic: Classical logic. Math. Struct. Comp. Sci. 1(3), 255–296 (1991)
8. Herbelin, H.: C'est maintenant qu'on calcule, au cœur de la dualité (2005), habilitation thesis
9. Jacobs, B.: Comprehension categories and the semantics of type dependency. Theor. Comput. Sci. 107(2), 169–207 (1993)
10. Lafont, Y., Reus, B., Streicher, T.: Continuation Semantics or Expressing Implication by Negation. Tech. rep., University of Munich (1993)
11. Laurent, O.: Etude de la polarisation en logique. Thèse de doctorat, Université Aix-Marseille II (mar 2002)
12. Laurent, O., Quatrini, M., Tortora de Falco, L.: Polarized and focalized linear and classical proofs. Ann. Pure Appl. Logic 134(2-3), 217–264 (2005)
13. Levy, P.B.: Call-by-Push-Value: A Subsuming Paradigm. In: Proc. TLCA '99. pp. 228–242 (1999)
14. Levy, P.B.: Adjunction models for call-by-push-value with stacks. In: Proc. Cat. Th. and Comp. Sci., ENTCS. vol. 69 (2005)
15. Loday, J.L.: Generalized bialgebras and triples of operads. arXiv preprint math/0611885 (2006), `http://arxiv.org/abs/math/0611885`
16. Melliès, P.A.: Asynchronous Games 3 An Innocent Model of Linear Logic. Electr. Notes Theor. Comput. Sci. 122, 171–192 (2005)
17. Melliès, P.A., Tabareau, N.: Resource modalities in tensor logic. Ann. Pure Appl. Logic 161(5), 632–653 (2010)
18. Moggi, E.: Computational Lambda-Calculus and Monads. In: LICS (1989)
19. Munch-Maccagnoni, G.: Focalisation and Classical Realisability. In: Proc. CSL '09. LNCS, Springer-Verlag (2009)
20. Munch-Maccagnoni, G.: Syntax and Models of a non-Associative Composition of Programs and Proofs. Ph.D. thesis, Univ. Paris Diderot (2013)
21. Murthy, C.R.: A Computational Analysis of Girard's Translation and LC. In: LICS. pp. 90–101. IEEE Computer Society (1992)
22. Selinger, P.: Re: co-exponential question. Message to the Category Theory mailing list (July 1999), `http://permalink.gmane.org/gmane.science.mathematics.categories/1181`
23. Selinger, P.: Control Categories and Duality: On the Categorical Semantics of the Lambda-Mu Calculus. Math. Struct in Comp. Sci. 11(2), 207–260 (2001)
24. Thielecke, H.: Categorical Structure of Continuation Passing Style. Ph.D. thesis, University of Edinburgh (1997)
25. Zeilberger, N.: On the unity of duality. Ann. Pure and App. Logic 153:1 (2008)