Operation through Enrichment

John C. Baez and Christian Williams

University of California, Riverside

14th May 2018

1 Introduction

Imagine you are making an airplane. This giant metal bird will carry humans, lovers and dreamers, high above the earth. Their lives rest entirely on the soundness of your design. How can you ensure their safety? You cannot - nature is fraught with accident; yet almost every day, we soar by the millions without a problem. This is physics, engineering, and testing; but there are limitations - we cannot simulate these constructions in every possible circumstance, and *prove* the security of our well-being.

The virtual world is another story. The computer scientist lacks the excuse of the airplane engineer: abstract thought is not burdened by the contingency of the physical world. Formal objects are defined and propositions are proven, once and for all - programs, languages, systems should be such objects. Yet so far, as we begin the connected era, this ideal is far from apparent. We've had bugs since the moth flew into the vacuum tube. Is there some basic limitation? No; only the inadequacy of unprincipled software development.

The problem: formal systems are often defined without intrinsic connection to how they actually *operate* in practice. In computation, the *structure* of the program is separate from the *dynamics* - but this disparity is the only source of error. If these are intertwined as one mathematical object, the system can be proven *correct by construction.* **Operational semantics** [12] is an essential tool in language design and verification, which formally specifies program behavior by *labelled transition systems*, or labelled directed graphs:

$$(\lambda x.x + x\ 2) \xrightarrow{\beta} 2 + 2 \xrightarrow{+} 4$$

The idea is to *reify* operational semantics via **enrichment** [5]: in the categorical representation of an algebraic theory, the objects are types, and the morphisms are terms. Thus to represent the actual *process* of computation, we need the higher-level notion of *rewriting* one term into another - the hom-object or "thing of morphisms" between two terms should be not a set but a *category*-like structure, where these 2-morphisms represent *rewrites.* For instance, the SK-combinator calculus is the "abstraction-free" λ-calculus:

$$\begin{tikzcd}[column sep=huge] 3 \arrow[d, bend right=50, "(((Sx)y)z)"'{name=L}] \arrow[d, bend left=50, "((xz)(yz))"{name=R}] & 2 \arrow[d, bend right=50, "((Kx)y)"'{name=L2}] \arrow[d, bend left=50, "x"{name=R2}] \\ 1 \arrow[from=L, to=R, Rightarrow, shorten=2pt] & 1 \arrow[from=L2, to=R2, Rightarrow, shorten=2pt] \end{tikzcd}$$

A **Lawvere theory** [7] defines an algebraic structure abstractly, as a category $\mathcal{T}$ generated by powers of a single object S and morphisms $S^n \to S$ representing n-ary operations, satisfying equations. This represents the *theory* of a kind of algebra, which can be modelled in a category $\mathcal{C}$ by a power-preserving functor $\mu : \mathcal{T} \to \mathcal{C}$. This is a very general notion of "algebra" - computational formalisms are also presented by generators and relations: in particular, a **term calculus** represents a *formal language* by sorts, term constructors, and congruence rules.

Enriched Lawvere theories for operational semantics has been explored in the past. It was studied in the case of categories by Seely [15], posets by Ghani and Lüth [10], and others, for various related purposes. Here, we follow the original idea of operational semantics with **graph-enriched** Lawvere theories:

sorts	: generating object S
term constructors	: generating morphisms $S^n \to S$
structural congruence	: commuting diagrams
$*$ rewrite rules	: generating hom-edges $*$

There are many other useful enriching categories. Better yet, there are functors between them that allow the seamless *transition* between different kinds of operational semantics. There is a *spectrum* of enriching categories which forms a gradient of resolution for the semantics of term calculi. For an enriching category $\mathcal{V}$, an $\mathbb{N}$-$\mathcal{V}$-theory is a $\mathcal{V}$-enriched Lawvere theory with $\mathbb{N}$-arities (see §4):

Graphs Gph-theories represent "small-step" operational semantics
- a hom-graph edge represents a *single* term rewrite.
Categories: Cat-theories represent "big-step" operational semantics:
- identity and composition represent the *reflexive-transitive* closure of the rewrite relation.
Posets: Pos-theories represent "full-step" operational semantics:
- a hom-poset boolean represents the *existence* of a big-step rewrite.
Sets: Set-theories represent denotational semantics:
- a hom-set element represents an *equivalence class* of the symmetric closure of the big-step relation.

Operational semantics is modelled and unified by enriched Lawvere theories and canonical functors between the enriching categories. This provides a more systematic categorical representation of computation.

A motivating example is "Logic as a distributive law" [16], an algorithm for deriving a spatial-behavioral type system from a formal presentation of a computational calculus. Not only can essential properties such as soundness be proven, but modalities can be "built-in" to express principles of the system.

This idea impacts more than computation, and though these concepts have been well-known for decades, the potential significance has not been fully realized. With the modern "proofs as programs" paradigm, this method applies to many subjects throughout mathematics. For example, one can construct the "theory of pseudomonoids" to model the rewrites of string diagrams...

2 Lawvere Theories

The "theory of monoids" can be defined without any reference to sets:

an object	M
an identity element	$e : 1 \to M$
and multiplication	$m : M^2 \to M$
with associativity	$m \circ (m \times M) = m \circ (M \times m)$
and unitality	$e \circ M = M = M \circ e$

Lawvere theories formalize this idea. They were originally called *finite product* theories: a skeleton $\mathbb{N}$ of the the category of finite sets FinSet is the free category with finite coproducts on 1 - every finite set is equal to the disjoint union of copies of $\{*\}$, and conversely $\mathbb{N}^{\mathrm{op}}$ is the free category with finite *products* on 1; so, a category with finite products $\mathcal{T}$ equipped with a strictly product-preserving bijective-on-objects functor $\iota : \mathbb{N}^{\mathrm{op}} \to \mathcal{T}$ is essentially a category generated by one object $\iota(1) = M$ and n-ary operations $M^n \to M$, as well as the projection and diagonal morphisms of finite products.

The abstraction of this definition is powerful: the syntax encapsulates the algebraic theory, *independent* of semantics, and then one is free to realize M as almost any mathematical object. For another category with finite products $\mathcal{C}$, a **model** of the Lawvere theory in $\mathcal{C}$ is a product-preserving functor $\mu : \mathcal{T} \to \mathcal{C}$. By the "free" property above, this functor is determined by $\mu(\iota(1)) = \mu(M) = X \in \mathcal{C}$. The models of $\mathcal{T}$ in $\mathcal{C}$ form a category $[\mathcal{T}, \mathcal{C}]_{fp}$, in which the morphisms are natural transformations. The general theory can be thereby modelled in many useful ways. For example, ordinary groups are models $\mathcal{T}_{\mathrm{Grp}} \to \mathrm{Set}$, but the theory can also be modelled in the category of topological spaces to form topological groups.

Lawvere theories and *finitary monads* provide complementary representations of algebraic structures and computation, as discussed by Hyland and Power in [4], and they have been proven equivalent by Linton in [8]. Let $\iota : \mathbb{N}^{\mathrm{op}} \to \mathcal{T}$ be a Lawvere theory and $\mathrm{Mod} = [\mathcal{T}, \mathrm{Set}]_{fp}$ be the category of models.

$$\begin{tikzcd}\mathrm{Set} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathrm{Mod} \arrow[l, bend left, "U"]\end{tikzcd}$$

There is the *underlying set* functor $U : \mathrm{Mod} \to \mathrm{Set}$ which sends each model $\mu : \mathcal{T} \to \mathrm{Set}$ to the image of the generating object, $\mu(\iota(1)) = X$ in Set. There is the *free model* functor $F : \mathrm{Set} \to \mathrm{Mod}$ which sends each finite set n to the representable $\mathcal{T}(n, -) : \mathcal{T} \to \mathrm{Set}$, and in general a set X to the set of all n-ary operations on it: $\{f(x_1, ..., x_n)|f \in \mathcal{T}(n, 1), x_i \in X\}$ - this is the *filtered colimit* of representables indexed by the poset of finite subsets of X [14], which pertains to conditions of finitude in §3. These form the adjunction:

$$\mathrm{Mod}(F(n), \mu) \cong \mu(n) \cong \mathrm{Set}(n, U(\mu))$$

The left isomorphism is by the Yoneda lemma, and the right isomorphism is by the definition of power in Set. Essentially, these are opposite ways of representing the n-ary operations of a given model. This adjunction induces a monad T on Set, which sends each set X to the set of all terms in the theory on X up to equality - the integral symbol is a *coend*, essentially a coproduct quotiented by the equality the theory:

$$T(X) = \int^{n \in \mathbb{N}} \mathcal{T}(n, 1) \times X^n$$

Conversely, for a monad T on Set, its *Kleisli category* is the category of all *free algebras* of the monad. There is a "comparison" functor Set $\to$ $Kl(T)$ which is the identity on objects and preserves products, so restricting to the subcategory of finite sets $\mathbb{N}$ forms the canonical Lawvere theory corresponding to the monad. This restriction is what limits the equivalence to *finitary* monads. There is a good explanation of all this in Milewski's categorical computation blog [11]. This generalizes to arbitrary *locally finitely presentable* modelling categories $\mathcal{C}$, which is discussed in §3.

The correspondence of Lawvere theories and finitary monads forms an equivalence of categories

$$Law \cong Mnd_f$$

as well as the categories of models and algebras for every corresponding pair $(\mathcal{T}, T)$:

$$\mathrm{Mod}(\mathcal{T}) \cong \mathrm{Alg}(T)$$

One motivation is to use monads is to define *distributive laws*, as utilized in [16]. The aforementioned references suffice; we do not need further details.

3 Enrichment

Let $(\mathcal{V}, \otimes, I)$ be a monoidal category [6]. A $\mathcal{V}$-enriched category or $\mathcal{V}$**-category** $\mathcal{C}$ is:

$$\begin{array}{rl}
\text{a collection of objects} & Obj(\mathcal{C}) \\
\text{a hom-object function} & \mathcal{C}(-,-) : Obj(\mathcal{C}) \times Obj(\mathcal{C}) \to Obj(\mathcal{V}) \\
\text{composition morphisms} & \circ_{a,b,c} : \mathcal{C}(b,c) \otimes \mathcal{C}(a,b) \to \mathcal{C}(a,c) \quad \forall a,b,c \in Obj(\mathcal{C}) \\
\text{identity elements} & i_a : I \to \mathcal{C}(a,a) \quad \forall a \in Obj(\mathcal{C})
\end{array}$$

such that composition is associative and unital. A $\mathcal{V}$**-functor** $F : \mathcal{C} \to \mathcal{D}$ is:

$$\begin{array}{rl}
\text{a function} & F_0 : Obj(\mathcal{C}) \to Obj(\mathcal{D}) \\
\text{hom-functions} & F_{ab} : \mathcal{C}(a,b) \to \mathcal{D}(Fa, Fb) \quad \forall a, b \in \mathcal{C}
\end{array}$$

such that F is compatible with composition and identity. A $\mathcal{V}$**-natural transformation** $\alpha : F \Rightarrow G$ is:

$$\text{a family} \quad \alpha_a : I \to \mathcal{D}(Fa, Ga) \quad \forall a \in Obj(\mathcal{C})$$

such that α is "natural" in a. See [5] for reference.

Let $\mathcal{V}$ be a **closed symmetric monoidal category**, providing

$$\begin{array}{rl} \text{internal hom} & [-,-] : \mathcal{V}^{\mathrm{op}} \otimes \mathcal{V} \to \mathcal{V} \\ \text{symmetry braiding} & \tau_{a,b} : a \otimes b \cong b \otimes a \quad \forall a, b \in Obj(\mathcal{C}) \\ \text{tensor-hom adjunction} & \mathcal{V}(a \otimes b, c) \cong \mathcal{V}(a, [b, c]) \quad \forall a, b, c \in Obj(\mathcal{V}) \end{array}$$

Then $\mathcal{V}$ is itself a $\mathcal{V}$-category, with internal hom as the hom-object function. The tensor-hom adjunction is very important: the counit $\varepsilon_{ab} : [a, b] \otimes a \to b$ is *evaluation*, in the fundamental sense. This adjunction generalizes to an *action* of $\mathcal{V}$ on any $\mathcal{V}$-category $\mathcal{C}$: for $x \in Obj(\mathcal{V})$ and $a, b \in Obj(\mathcal{C})$, the **power** of b by x and the **copower** of a by x are objects of $\mathcal{C}$ which represent the adjunction:

$$\mathcal{C}(a \odot x, b) \cong \mathcal{V}(x, \mathcal{C}(a, b)) \cong \mathcal{C}(a, x \pitchfork b)$$

and $\mathcal{C}$ is $\mathcal{V}$-powered or copowered if all powers or copowers exist.

These are the two basic forms of enriched limit and colimit, which are not especially intuitive; but they are a direct generalization of a familiar idea in the category of sets. In Set, the power is the "exponential" function set and the copower is the product. To generalize this to an action on other Set-categories, note:

$$X \pitchfork Y = \quad X^Y \quad \cong \textstyle\prod_{y \in Y} X$$

$$X \odot Y = \quad X \times Y \quad \cong \textstyle\coprod_{y \in Y} X$$

So, categories are canonically Set-powered or copowered by indexed products or coproducts of copies of an object, provided that these exist. So in the definition of Lawvere theory, even though it seems to be all about products, it is actually about *powers*, because these constitute the *arities* of the operations. This is precisely what is generalized in the enriched form. We will use exponential notation $x \pitchfork b = b^x$ for simplicity, and because the enriching categories under consideration are cartesian.

There are just a few more technicalities. * Given a $\mathcal{V}$-category $\mathcal{C}$, one often considers the Yoneda embedding into the $\mathcal{V}$-presheaf category $[\mathcal{C}^{\mathrm{op}}, \mathcal{V}]$, and it is important if certain subcategories are representable; generally, some properties of $\mathcal{C}$ depend on a condition of "finitude." [1] A category is **locally finitely presentable** if it is the category of models for a *sketch*, which is a generalization of Lawvere theory to finite limits, and an object is finitely presentable or **finite** if its representable functor is *finitary*, or preserves filtered colimits. A $\mathcal{V}$-category $\mathcal{C}$ is locally finitely presentable if the underlying category $\mathcal{C}_0$ is LFP, $\mathcal{C}$ has finite powers, and $(-)^x : \mathcal{C}_0 \to \mathcal{C}_0$ is finitary. The details are not crucial - all categories to be considered are locally finitely presentable. Denote by $\mathcal{V}_f$ the subcategory of $\mathcal{V}$ of finite objects - in Gph, these are simply graphs with finite vertices and edges.

4 $\mathcal{J}$-$\mathcal{V}$ theories

All of these abstract definitions culminate in the central concept: for a symmetric monoidal closed category $(\mathcal{V}, \otimes, I)$, a *$\mathcal{V}$-enriched Lawvere theory* à la Power [13] is a finitely-powered $\mathcal{V}$-category $\mathcal{T}$ equipped with a strictly power-preserving bijective-on-objects $\mathcal{V}$-functor $\iota : \mathcal{V}_f^{\mathrm{op}} \to \mathcal{T}$. A *model* of a $\mathcal{V}$-theory is a finite-power $\mathcal{V}$-functor $\mu : \mathcal{T} \to \mathcal{V}$, and $\mathcal{V}$-natural transformations between them form the $\mathcal{V}$-category of models $[\mathcal{T}, \mathcal{V}]_{fp}$. The monadic adjunction and equivalence of §2 generalize to $\mathcal{V}$-theories, as originally formulated by Power in [13].

However, this requires $\mathcal{T}$ to have all powers of $\mathcal{V}_f$, i.e. the theory must have arities for every finite object of $\mathcal{V}$. It is certainly useful to include these generalized arities, but this introduces the question of how to *present* such a theory; this is not nearly as straightforward as n-ary operations, and to the authors' knowledge a general method of enriched presentation does not yet exist. However, this is not needed for our purposes - we only need *natural number* arities, while still retaining graph-enrichment. How can this be defined?

A very general and useful definition of enriched algebraic theory was introduced by Lucyshyn-Wright [9], which allows for theories to be parameterized by a **system of arities**, a full subcategory inclusion $j : \mathcal{J} \hookrightarrow \mathcal{V}$ containing the monoidal unit and closed under tensor.

A $\mathcal{V}$-enriched algebraic theory with j-arities or **$\mathcal{J}$-$\mathcal{V}$ theory** is a $\mathcal{V}$-category $\mathcal{T}$ equipped with a $\mathcal{J}$-power preserving bijective-on-objects $\mathcal{V}$-functor $\tau : \mathcal{J}^{\mathrm{op}} \to \mathcal{T}$. This is essentially a $\mathcal{V}$-category with objects being $\mathcal{J}$-powers of a generating object $S = S^I$, S^J for $J \in \mathcal{J}$. A **model** of this theory in a $\mathcal{V}$-category $\mathcal{C}$ is a finite-power preserving $\mathcal{V}$-functor $\mathcal{T} \to \mathcal{C}$. Here is an overview of the concepts:

$$\begin{array}{ccccc}
j : & \mathcal{J} & \hookrightarrow & \mathcal{V} & \text{arities} \\
 & & & \frown & \text{enrichment} \\
\tau : & \mathcal{J}^{\mathrm{op}} & \to & \mathcal{T} & \text{theory} \\
 & & & (\downarrow) & \text{models} \\
 & & & \mathcal{C} & \text{semantics}
\end{array}$$

This subsumes existing formulations; for example, Power's definition is the case $\mathcal{J} = \mathcal{V}_f$. A system of arities is **eleutheric** if left Kan extensions along j exist and are preserved by $\mathcal{V}(K, -)$ for all $K \in Ob(\mathcal{J})$. This is what is needed to have the essential *monadicity* theorems: Lucyshyn-Wright proved that any $\mathcal{J}$-$\mathcal{V}$ theory for an eleutheric system of arities has a category of models for $\mathcal{C} = \mathcal{V}$ which is monadic over $\mathcal{V}$, and the induced $\mathcal{V}$-monad is "j-ary" in that it "conditionally preserves $\mathcal{J}$-flat colimits, i.e. can be thought of as a monad with $\mathcal{J}$ arities.

The usual kinds of arities were all proved to be eleutheric: in particular, finite cardinals. The structure described in the introduction is precisely an $\mathbb{N}$-Gph-theory.

5 Change of Base

We propose a general framework in which one can *transition* seamlessly between different forms of operational semantics. This is effected by a **monoidal functor** - a functor

$$F : (\mathcal{V}, \otimes_{\mathcal{V}}, I_{\mathcal{V}}) \to (\mathcal{W}, \otimes_{\mathcal{W}}, I_{\mathcal{W}})$$

which transfers the tensor and unit via the *laxor* and *unitor*

$$\begin{array}{ll}
\lambda : & F(a) \otimes_{\mathcal{W}} F(b) \to F(a \otimes_{\mathcal{V}} b) \\
\upsilon : & I_{\mathcal{W}} \to F(I_{\mathcal{V}})
\end{array}$$

This induces a **change of base** functor $\varphi : \mathcal{V}\mathrm{Cat} \to \mathcal{W}\mathrm{Cat}$ [2]. Suppose

$$\beta \circ \alpha : P \Rightarrow Q \Rightarrow R : \mathcal{C} \to \mathcal{D} \in \mathcal{V}\mathrm{Cat}$$

Then φ is defined:

$$\begin{array}{rlrl}
\varphi(\mathcal{C}) \in \mathcal{W}\mathrm{Cat} & & \varphi(\mathcal{V}\mathrm{Cat}(\mathcal{C}, \mathcal{D})) \in \mathcal{W} & \\
 & & & \\
\text{objects} & Obj(\mathcal{C}) & \text{objects} & \varphi(G)_{obj} = G,\ \varphi(G)_{hom} = F \circ G \\
\text{hom-function} & F \circ \mathcal{C}(-, -) & \text{hom-function} & \varphi(\alpha) = F(\alpha) \circ \upsilon \\
\text{composition} & F(\circ_{a,b,c}) \circ \lambda & \text{composition} & \varphi(\beta \circ \alpha) = \varphi(\beta) \circ \varphi(\alpha) \\
\text{identity} & F(i_a) \circ \upsilon & \text{identity} & \varphi(1_G) = 1_{\varphi(G)}
\end{array}$$

A monoidal functor $F : \mathcal{V} \to \mathcal{W}$ with change of base $\varphi : \mathcal{V}\mathrm{Cat} \to \mathcal{W}\mathrm{Cat}$ is a **change of semantics** for a $\mathcal{J}$-$\mathcal{V}$ theory (j, τ) if $(F \circ j, \varphi(\tau))$ is a $F(\mathcal{J})$-$\mathcal{W}$ theory. This requires $\varphi(j)$ to be full and $\varphi(\tau)$ to be bijective-on-objects and $F(J)$-power preserving.

Note: in the same way that every $n \in \mathbb{N}^{\mathrm{op}}$ is a power of $1 \in \mathrm{Set}$, every $J \in \mathcal{J}$ is a power of the monoidal unit $I \in \mathcal{V}$:

$$\mathcal{J}(a \odot I, b) \cong \mathcal{V}(I, \mathcal{J}(a, b)) \cong \mathcal{J}(a, I \pitchfork b)$$

- substituting a for b, the unital tensor or 'copower' isomorphism on the left maps to the unital 'cotensor' or power isomorphism. Since a $\mathcal{J}$-$\mathcal{V}$ theory preserves powers, this implies that every object of $\mathcal{T}$ is a power of the unit.

Theorem. If F is full and power-preserving, then F is a change of semantics.

Proof. If F is full, then $F \circ j$ is full. By the above note, if F preserves powers, then $F(J) = F([I, J]) = [F(I), F(J)]$, and these determine the action of $\varphi(\tau)$ by $F(J)$-powers of $F(I)$. □

6 Semantics

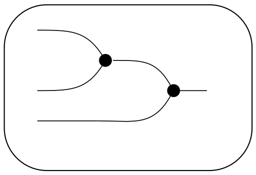
$$\begin{tikzcd}
\mathrm{Gph} \arrow[r, bend left, "F_C"] \arrow[r, phantom, "\perp"] & \mathrm{Cat} \arrow[l, bend left, "U_G"] \arrow[r, bend left, "F_P"] \arrow[r, phantom, "\perp"] & \mathrm{Pos} \arrow[l, bend left, "U_C"] \arrow[r, bend left, "U_S"] \arrow[r, phantom, "\top"] & \mathrm{Set} \arrow[l, bend left, "F_P"]
\end{tikzcd}$$

7 Application

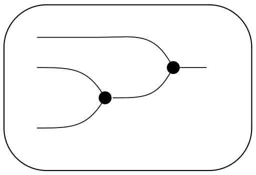
Pseudomonoids [3] are important in higher category theory. A monoidal category is "just a pseudomonoid in Cat!" The tensor is only associative and unital *up to isomorphism*; these are now reified as 2-morphisms in a Cat-enriched Lawvere theory, and can be understood as invertible rewrites of string diagrams:

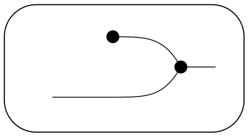
$$\boxed{\mathrm{Th(PsMon)}}$$

$$M^2 \xrightarrow{m} M \xleftarrow{e} 1$$

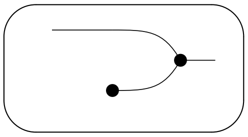


$$\begin{tikzcd}
M^3 \arrow[r, "M\times m"] \arrow[d, "m\times M"'] \arrow[dr, phantom, "\overset{a}{\Longrightarrow}"] & M^2 \arrow[d, "m"] \\
M^2 \arrow[r, "m"'] & M
\end{tikzcd}$$





$$\begin{tikzcd}
M \arrow[r, "e\times M"] \arrow[d, "M\times e"'] \arrow[dr, equal] \arrow[dr, phantom, "l", shift left=3] \arrow[dr, phantom, "r"', shift right=3] & M^2 \arrow[d, "m"] \\
M^2 \arrow[r, "m"'] & M
\end{tikzcd}$$



$$\begin{tikzcd}
M^4 \arrow[rr, "M\times M\times m"] \arrow[dr, "M\times m\times M"'] \arrow[dd, "m\times M\times M"'] & & M^3 \arrow[dr, "M\times m"] \arrow[d, phantom, "\overset{M\times a}{\Rrightarrow}"] & \\
& M^3 \arrow[rr, "M\times m"] \arrow[dd, "m\times M"] \arrow[dl, phantom, "a\times M\ \Rrightarrow"] & & M^2 \arrow[dd, "m"] \\
M^3 \arrow[dr, "m\times M"'] & & \overset{a}{\Rrightarrow} & \\
& M^2 \arrow[rr, "m"] & & M
\end{tikzcd}$$

$$\begin{tikzcd}
M^4 \arrow[rr, "M\times M\times m"] \arrow[dd, "m\times M\times M"'] & & M^3 \arrow[dr, "M\times m"] \arrow[dd, "m\times M"'] & \\
& \overset{?}{\Rrightarrow} & & M^2 \arrow[dd, "m"] \\
M^3 \arrow[rr, "M\times m"] \arrow[dr, "m\times M"'] & & M^2 \arrow[dr, "m"] \arrow[ur, phantom, "a\ \Rrightarrow"] & \\
& M^2 \arrow[rr, "m"] \arrow[ur, phantom, "\underset{a}{\Rrightarrow}"] & & M
\end{tikzcd}$$

References

[1] J. Adamek and J. Rosicky, *Locally presentable and accessible categories*, London Mathematical Society Lecture Notes Series **189** (1994).

[2] Francis Borceux, *Handbook of cateogorical algebra*, vol. 2, 1994.

[3] Brian Day and Ross Street, *Monoidal categories and hopf algebroids*, Advances in Mathematics **129** (1997).

[4] Martin Hyland and John Power, *The category theoretic understanding of universal algebra: Lawvere theories and monads*, Electronic Notes in Theoretical Computer Science **172** (2007).

[5] G.M. Kelly, *Basic concepts of enriched category theory*, vol. 64, 1982.

[6] Saunders Mac Lane, *Categories for the working mathematician*, Graduate Texts in Mathematics (1971).

[7] F. William Lawvere, *Functorial semantics of algebraic theories*, Reports of the Midwest Category Seminar II **5** (2004).

[8] F.E.J. Linton, *Some aspects of equational theories*, Proceedings of the Conference on Categorical Algebra (1965).

[9] Rory B.B. Lucyshyn-Wright, *Enriched algebraic theories and monads for a system of arities*, Logic in Computer Science (2016).

[10] Christoph Lüth and Neil Ghani, *Monads and modular term rewriting*, Lecture Notes in Computer Science (1997).

[11] Bartosz Milewski, *Lawvere theories*, https://bartoszmilewski.com/2017/08/26/lawvere-theories/, 2017.

[12] Gordon D. Plotkin, *A strucural approach to operational semantics*, Journal of Logic and Algebraic Programming (1981).

[13] John Power, *Enriched lawvere theories*, Theory and Applications of Categories (2000).

[14] Urs Schreiber, *Lawvere theory*, (2010).

[15] R.A.G. Seely, *Modelling computations: A 2-categorical framework*, Symposium on Logic in Computer Science (1987).

[16] Michael Stay and L.G. Meredith, *Logic as a distributive law*, Logic in Computer Science (2016).