ENRICHED LAWVERE THEORIES FOR OPERATIONAL SEMANTICS

1. Introduction

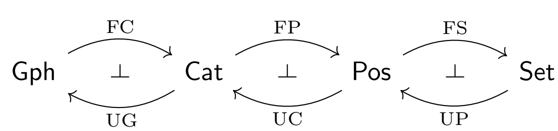
Formal systems are not always explicitly connected to how they operate in practice. Lawvere theories [18] are an excellent formalism for describing algebraic structures obeying equational laws, but they do not specify how to compute in such a structure, for example taking a complex expression and simplifying it using rewrite rules.

In a Lawvere theory the objects are types and the morphisms are terms; however there are no relations between terms, only equations. The process of computing one term into another should be given by hom-objects with more structure. In operational semantics, program behavior is often specified by labelled transition systems, or labelled directed graphs [28]. The edges of such a graph represent rewrites. We can use an enhanced Lawvere theory in which, rather than merely *sets* of morphisms, there are *graphs* or perhaps *categories*.

To be clear, this is certainly not a new idea. Using enriched Lawvere theories for operational semantics has been explored in the past. For example, category-enriched theories have been studied by Seely [32] for the λ-calculus, and poset-enriched ones by Ghani and Lüth [21] for understanding "modularity" in term rewriting systems. They have been utilized extensively by Power, enriching in ω-complete partial orders to study recursion [29] – in fact, there the simplified "natural number" enriched theories which we explore were implicitly considered.

In the context of these works, the purpose of the present paper is to provide a simple, general exposition of enriched theories: we hope to familiarize computer scientists with enriched category theory, and prove some basic results to show that one does not need to leave the nice computational world of cartesian closed categories to enjoy the benefits of enrichment.

For an enriching category V, we take a **V-theory** to be a V-enriched Lawvere theory with natural number arities. There is a "spectrum" of enriching categories which allow us to examine the semantics of term calculi at various levels of detail. We discuss how functors between enriching categories induce change-of-base 2-functors between their 2-categories of enriched categories, and we show that functors preserving finite products induce *change-of-semantics*: that is, they map theories to theories and models to models. Our main examples arise from this chain of adjunctions:



The right adjoints here automatically preserve finite products, but the left adjoints do as well, and these are more important in applications:

Change of base along FC maps small-step to big-step operational semantics.
Change of base along FP maps big-step to full-step operational semantics.
Change of base along FS maps full-step operational semantics to denotational semantics.

Acknowledgements. This paper builds upon the ideas of Mike Stay and Greg Meredith presented in "Representing operational semantics with enriched Lawvere theories" [35]. We appreciate their offer to let us develop this work further for use in the innovative distributed computing system RChain, and gratefully acknowledge the support of Pyrofex Corporation. We also thank Richard Garner, Todd Trimble and others at the n-Category Café for classifying cartesian closed categories where every object is a finite coproduct of copies of the terminal object [2].

2. Lawvere Theories

Algebraic structures are traditionally treated as sets equipped with operations obeying equations, but we can generalize such structures to live in any category with finite products. For example, given any category C with finite products, we can define a monoid internal to C to consist of:

an object	M
an identity element	$e\colon 1 \to M$
and multiplication	$m\colon M^2 \to M$
obeying the associative law	$m \circ (m \times M) = m \circ (M \times m)$
and the right and left unit laws	$m \circ (e \times \mathrm{id}_M) = \mathrm{id}_M = m \circ (\mathrm{id}_M \times e)$.

Lawvere theories formalize this idea. For example, there is a Lawvere theory $\mathsf{Th}(\mathsf{Mon})$, the category with finite products freely generated by an object t equipped with an identity element $e\colon 1 \to t$ and multiplication $m\colon t^2 \to t$ obeying the associative law and unit laws listed above. This captures the "Platonic idea" of a monoid internal to a category with finite products. A monoid internal to C then corresponds to a functor $\mu\colon \mathsf{T} \to \mathsf{C}$ that preserves finite products.

In more detail, let N be any skeleton of the category of finite sets FinSet. Because N is the free category with finite coproducts on 1, N^{op} is the free category with finite products on 1. A **Lawvere theory** is a category with finite products T equipped with a functor $\tau\colon \mathsf{N}^{\mathrm{op}} \to \mathsf{T}$ that is bijective on objects and preserves finite products. Thus, a Lawvere theory is essentially a category generated by one object $\tau(1) = t$ and n-ary operations $t^n \to t$, as well as the projection and diagonal morphisms of finite products.

For efficiency let us call a functor that preserves finite products **cartesian**. Lawvere theories are the objects of a category Law whose morphisms are cartesian functors $f\colon \mathsf{T} \to \mathsf{T}'$ that obey $f\tau = \tau'$. More generally, for any category with finite products C, a **model** of the Lawvere theory T in C is a cartesian functor $\mu\colon \mathsf{T} \to \mathsf{C}$. The models of T in C are the objects of a category $\mathsf{Mod}(\mathsf{T}, \mathsf{C})$, in which the morphisms are natural transformations.

Making a Lawvere theory from a "sketch" of operations and equations is just like the presentation of an algebra by generators and relations: we form the free category with finite products on the data given, and impose the required equations. The result is a category whose objects are powers of M, and whose morphisms are composites of products of the morphisms in $\mathsf{Th}(\mathsf{Mon})$, projections, deletions, symmetries and diagonals. In §4 we see that this construction is actually given by the Cat-theory for categories with finite products.

3. Enrichment

To allow more general semantics, we now turn to Lawvere theories that have hom-*objects* rather than mere hom-*sets*. To do this we use enriched category theory [16] and replace sets with objects of a cartesian closed category V, called the "enriching" category or "base". A V-enriched category or V**-category** C is:

$$\begin{array}{rl} \text{a collection of objects} & \mathrm{Ob}(\mathsf{C}) \\ \text{a hom-object function} & \mathsf{C}(-,-)\colon \mathrm{Ob}(\mathsf{C}) \times \mathrm{Ob}(\mathsf{C}) \to \mathrm{Ob}(\mathsf{V}) \\ \text{composition morphisms} & \circ_{a,b,c}\colon \mathsf{C}(b,c) \times \mathsf{C}(a,b) \to \mathsf{C}(a,c) \quad \forall a,b,c \in \mathrm{Ob}(\mathsf{C}) \\ \text{identity-assigning morphisms} & i_a\colon 1_{\mathsf{V}} \to \mathsf{C}(a,a) \quad \forall a \in \mathrm{Ob}(\mathsf{C}) \end{array}$$

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

$$\begin{array}{rl} \text{a function} & F\colon \mathrm{Ob}(\mathsf{C}) \to \mathrm{Ob}(\mathsf{D}) \\ \text{a collection of morphisms} & F_{ab}\colon \mathsf{C}(a,b) \to \mathsf{D}(F(a),F(b)) \quad \forall a,b \in \mathsf{C} \end{array}$$

such that F preserves composition and identity. A V**-natural transformation** $\alpha\colon F \Rightarrow G$ is:

$$\text{a family} \quad \alpha_a\colon 1_{\mathsf{V}} \to \mathsf{D}(F(a),G(a)) \quad \forall a \in \mathrm{Ob}(\mathsf{C})$$

such that α is "natural" in a. There is a 2-category VCat of V-categories, V-functors, and V-natural transformations.

We can construct new V-categories from old by taking products and opposites in an obvious way. There is also a V-category denoted $\underline{\mathsf{V}}$ with the same objects as V and with hom-objects given by the internal hom:

$$\underline{\mathsf{V}}(v,w) = w^v \quad \forall v,w \in \mathsf{V}.$$

We can generalize products and coproducts to the enriched context. Given a V-category C, a V**-product** of an n-tuple of objects $b_1,\ldots,b_n \in \mathrm{Ob}(C)$ is an object b equipped with V-natural isomorphism

$$\mathsf{C}(-,b) \cong \prod_{i=1}^{n} \mathsf{C}(-,b_i). \tag{1}$$

If such an object b exists, we denote it by $\prod_{i=1}^n b_i$. This makes sense even when $n = 0$: a 0-ary product in C is called a V**-terminal object** and denoted as 1_{C}.

Whenever V is cartesian closed, the finite products in V are also V-products in $\underline{\mathsf{V}}$; this mainly amounts to saying

$$(u \times v)^w \cong u^w \times v^w \quad \text{and} \quad 1_{\mathsf{V}}^w \cong 1_{\mathsf{V}}.$$

Conversely, any finite V-product in V is also a product in the usual sense. In a general V-category C, it makes no sense to say a V-product is a product in the usual sense. However, the V-natural isomorphism in Eq. (1) gives rise to a morphism

$$\pi_i\colon 1_{\mathsf{V}} \to \mathsf{C}(b,b_i)$$

for each i, defined as the composite

$$1_{\mathsf{V}} \xrightarrow{i_b} \mathsf{C}(b,b) \xrightarrow{\sim} \prod_{i=1}^{n} \mathsf{C}(b,b_i) \longrightarrow \mathsf{C}(b,b_i).$$

These morphisms π_i serve as substitutes for the projections from b to b_i. They are "elements" of the hom-objects $\mathsf{C}(b,b_i)$, where an **element** of $v \in \mathrm{Ob}(\mathsf{V})$ is a morphism from 1_{V} to v. Elements of hom-objects behave much like morphisms for C. For example, we can define a morphism

$$p_i\colon \mathsf{C}(a,b) \to \mathsf{C}(a,b_i),$$

which acts like composition with π_i as follows:

$$\mathsf{C}(a,b) \xrightarrow{\sim} 1_{\mathsf{V}} \times \mathsf{C}(a,b) \xrightarrow{\pi_i \times 1} \mathsf{C}(b,b_i) \times \mathsf{C}(a,b) \xrightarrow{\circ_{a,b,b_i}} \mathsf{C}(a,b_i).$$

This morphism is V-natural in a, and the isomorphism in Eq. (1) has components given by the morphisms p_i.

We say that a V-functor $F\colon \mathsf{C} \to \mathsf{D}$ **preserves** V-products if for every $b = \prod_{i=1}^n b_i$ in C, the V-natural transformations

$$F(p_i)\colon \mathsf{D}(-, F(b)) \to \mathsf{D}(-, F(b_i))$$

are the components of a V-natural isomorphism

$$\mathsf{D}(-, F(b)) \cong \prod_{i=1}^n \mathsf{D}(-, F(b_i)),$$

and similarly for V-coproducts.

A bit more subtly, generalizing the product and internal hom of V, a V-category C can have "tensors" and "powers" (which are sometimes called "copowers" and "cotensors"). Given $a \in \mathrm{Ob}(\mathsf{C})$ and $v \in \mathrm{Ob}(\mathsf{V})$, we say an object $v \cdot a \in \mathrm{Ob}(\mathsf{C})$ is the **tensor** of a by v if it is equipped with isomorphisms

$$\mathsf{C}(v \cdot a, b) \cong \mathsf{C}(a, b)^v$$

V-natural in b. In the special case $\mathsf{V} = \mathsf{Set}$ this forces $v \cdot a$ to be the v-fold coproduct of copies of a:

$$v \cdot a = \sum_{i \in v} a.$$

Similarly, given $b \in \mathrm{Ob}(\mathsf{C})$ and $v \in \mathrm{Ob}(\mathsf{V})$, we say an object $b^v \in \mathrm{Ob}(C)$ is a **power** of b by v if it is equipped with isomorphisms

$$\mathsf{C}(a, b^v) \cong \mathsf{C}(a, b)^v$$

V-natural in a. In the special case $\mathsf{V} = \mathsf{Set}$ this forces b^v to be the v-fold product of copies of b:

$$b^v = \prod_{i \in v} b.$$

As with V-products, the V-natural isomorphism of powers gives rise to "projections" given by

$$\varepsilon_{b,v}\colon 1_{\mathsf{V}} \xrightarrow{i_{b^v}} \mathsf{C}(b^v, b^v) \xrightarrow{\sim} \mathsf{V}(v, \mathsf{C}(b^v, b))$$

and these allow us to define "internal projections"

$$e_{b,v}\colon \mathsf{C}(a, b^v) \to \mathsf{V}(v, \mathsf{C}(a, b))$$

given by

$$\mathsf{C}(a, b^v) \xrightarrow{\sim} 1_{\mathsf{V}} \times \mathsf{C}(a, b^v) \xrightarrow{\epsilon_{b,v} \times 1} \mathsf{V}(v, \mathsf{C}(b^v, b)) \times \mathsf{C}(a, b^v) \xrightarrow{\sim}$$

$$\mathsf{V}(v, \mathsf{C}(b^v, b)) \times \mathsf{V}(1_{\mathsf{V}}, \mathsf{C}(a, b^v)) \xrightarrow{\times} \mathsf{V}(v \times 1_{\mathsf{V}}, \mathsf{C}(b^v, b) \times \mathsf{C}(a, b^v)) \xrightarrow{\circ_{a,b^v,b}} \mathsf{V}(v, \mathsf{C}(a, b))$$

We say that a V-functor $F\colon \mathsf{C} \to \mathsf{D}$ **preserves powers** if for every b in C and v in V, the V-natural transformations

$$F(e_{b,v})\colon \mathsf{D}(-, F(b^v)) \to \mathsf{V}(v, \mathsf{C}(-, F(b)))$$

are the components of a V-natural isomorphism

$$\mathsf{D}(-, F(b^v)) \cong \mathsf{V}(v, \mathsf{D}(-, F(b))),$$

and similarly for tensors.

4. Enriched Lawvere Theories

Power introduced the notion of enriched Lawvere theory about twenty years ago, "in seeking a general account of what have been called notions of computation" [30]. The original definition is as follows: for a symmetric monoidal closed category $(\mathsf{V}, \otimes, 1)$, a "V-enriched Lawvere theory" is a V-category T that has powers by objects in V_f, equipped with an identity-on-objects V-functor

$$\tau\colon \underline{\mathsf{V}}_f^{\mathrm{op}} \to \mathsf{T}$$

that preserves these powers. A "model" of a V-theory is a V-functor $\mu\colon \mathsf{T} \to \mathsf{V}$ that preserves powers by finite objects of V. There is a category $\mathsf{Mod}(\mathsf{T}, \mathsf{V})$ whose objects are models and whose morphisms are V-natural transformations. However, this sort of V-enriched Lawvere theory has arities for every finite object of V. In this paper, however, we only consider *natural number* arities, while still retaining enrichment. To do this we use the work of Lucyshyn-Wright [20], who along with Power [27] has generalized Power's original ideas to allow a more flexible choice of arities. We also limit ourselves to the case where the tensor product of V is cartesian. This has a significant simplifying effect, yet it suffices for many cases of interest in computer science.

Thus, in all that follows, we let $(\mathsf{V}, \times, 1_{\mathsf{V}})$ be a cartesian closed category equipped with chosen finite coproducts of the terminal object 1_{V}, say

$$n_{\mathsf{V}} = \sum_{i \in n} 1_{\mathsf{V}}.$$

Define N_{V} to be the full subcategory of V containing just these objects n_{V}. There is also a V-category $\underline{\mathsf{N}}_{\mathsf{V}}$ whose objects are those of N_{V} and whose hom-objects are given as in V. We define the V-category of **arities** for V to be

$$\mathsf{A}_{\mathsf{V}} := \underline{\mathsf{N}}_{\mathsf{V}}^{\mathrm{op}}.$$

We shall soon see that A_{V} has finite V-products.

Definition 1. We define a V-**theory** (T, τ) to be a V-category T equipped with a V-functor

$$\tau\colon \mathsf{A}_{\mathsf{V}} \to \mathsf{T}$$

that is bijective on objects and preserves finite V-products.

Definition 2. A **model** of T in a V-category C is a V-functor

$$\mu\colon \mathsf{T} \to \mathsf{C}$$

that preserves finite V-products.

Just as all the objects of a Lawvere theory are finite products of a single object, we shall see that all the objects of T are finite V-products of the object

$$t = \tau(1_{\mathsf{V}}).$$

Definition 3. For every V-theory (T, τ) and every V-category C with finite V-products, we define $\mathsf{Mod}(\mathsf{T}, \mathsf{C})$, the **category of models** of (T, τ) in C, to be the category for which an object is a V-functor $\mu\colon \mathsf{T} \to \mathsf{C}$ that preserves finite V-products and a morphism is a V-natural transformation.

Definition 4. We define VLaw, the **category of** V-**theories**, to be the category for which an object is a V-theory and a morphism from (T, τ) to (T', τ') is a V-functor $f\colon \mathsf{T} \to \mathsf{T}'$ that preserves finite V-products and has $f\tau = \tau'$.

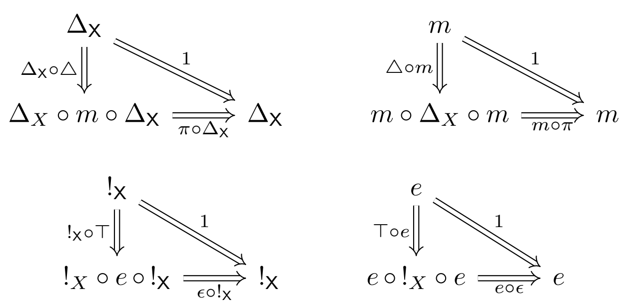
Example 5. Enrichment generalizes operations in more ways than by weakening equations to coherent isomorphisms. We can also use 2-theories to describe other structures that make sense inside 2-categories, such as adjunctions.

For example, we may define a cartesian category X to be one equipped with right adjoints to the diagonal $\Delta_{\mathsf{X}} \colon \mathsf{X} \to \mathsf{X} \times \mathsf{X}$ and the unique functor $!_{\mathsf{X}} \colon \mathsf{X} \to 1_{\mathsf{Cat}}$. These right adjoints are a functor $m \colon \mathsf{X}^2 \to \mathsf{X}$ describing binary products in X and a functor $e \colon 1 \to \mathsf{X}$ picking out the terminal object in X. We can capture the fact that they are right adjoints by providing them with units and counits and imposing the triangle equations. There is thus a 2-theory $\mathsf{Th}(\mathsf{Cart})$ whose models in Cat are categories with chosen finite products. More generally a model of this 2-theory in any 2-category C with finite products is called a **cartesian object** in C.

$\mathsf{Th}(\mathsf{Cart})$

type	X	cartesian object
operations	$m \colon \mathsf{X}^2 \to \mathsf{X}$	product
	$e \colon 1 \to \mathsf{X}$	terminal element
rewrites	$\triangle \colon \mathrm{id}_{\mathsf{X}} \Longrightarrow m \circ \Delta_{\mathsf{X}}$	unit of adjunction between m and Δ_{X}
	$\pi \colon \Delta_{\mathsf{X}} \circ m \Longrightarrow \mathrm{id}_{\mathsf{X}^2}$	counit of adjunction between m and Δ_{X}
	$\top \colon \mathrm{id}_{\mathsf{X}} \Longrightarrow e \circ !_{\mathsf{X}}$	unit of adjunction between e and $!_{\mathsf{X}}$
	$\epsilon \colon !_{\mathsf{X}} \circ e \Longrightarrow \mathrm{id}_1$	counit of adjunction between e and $!_{\mathsf{X}}$

equations



Again we write the equations as commutative diagrams, but this time commutative triangles of 2-morphisms in $\mathsf{Th}(\mathsf{Cart})$. These are the triangle equations that force m to be the right adjoint of Δ_{X} and e to be the right adjoint of $!_{\mathsf{X}}$. A model of $\mathsf{Th}(\mathsf{Cart})$ is a category with chosen binary products and a chosen terminal object; morphisms in $\mathsf{Mod}(\mathsf{Th}(\mathsf{Cart}), \mathsf{Cat})$ are functors that strictly preserve this extra structure.

In fact, if we let arities be finite categories, we would have Cat-theories of categories with finite limits and colimits. However, for the purposes of this paper we are using only natural number arities. This suffices for constructing $\mathsf{Th}(\mathsf{Cart})$ and also $\mathsf{Th}(\mathsf{CoCart})$, the theory of categories with chosen binary coproducts and a chosen initial object. Various other kinds of categories—distributive categories, rig categories, etc.—can also be expressed using Cat-theories with natural number arities. This gives a systematic formalization of these categories, internalizes them to new contexts, and allows for the generation of 2-monads that describe them.

5. Natural Number Arities

Lemma 6. Let V be cartesian closed with chosen finite coproducts of the terminal object and let T be a V-category. These conditions for a V-functor $\tau\colon \mathsf{A}_\mathsf{V} \to \mathsf{T}$ are equivalent:

(1) (T, τ) is a V-theory,
(2) τ preserves finite V-products,
(3) τ preserves powers by objects of N_V.

6. Change of Base

We now have the tools to formulate the main idea: a choice of enrichment for Lawvere theories corresponds to a choice of *semantics*, and changing enrichments corresponds to a *change of semantics*. We propose a general framework in which one can translate between different forms of semantics: small-step, big-step, full-step operational semantics, and denotational semantics.

Suppose that V and W are enriching categories of the sort we are considering: cartesian closed categories equipped with chosen finite coproducts of the terminal object. Suppose $F\colon \mathsf{V} \to W$ preserves finite products. This induces a **change of base** functor $F_*\colon \mathsf{VCat} \to \mathsf{WCat}$ [9] which takes any V-category C and produces a W-category $F_*(\mathsf{C})$ with the same objects but with

$$F_*(\mathsf{C})(a, b) := F(\mathsf{C}(a, b))$$

for all objects a, b. Composition in $F_*(\mathsf{C})$ is defined by

$$F(\mathsf{C}(b, c)) \times F(\mathsf{C}(a, b)) \xrightarrow{\sim} F(\mathsf{C}(b, c) \times \mathsf{C}(a, b)) \xrightarrow{F(\circ_{a,b,c})} F(C(a, b)).$$

The identity-assigning morphisms are given by

$$1 \xrightarrow{\sim} F(1) \xrightarrow{F(i_a)} F(\mathsf{C}(a, b)).$$

Moreover, if $f\colon \mathsf{C} \to \mathsf{D} \in \mathsf{VCat}$ is a V-functor, there is a W-functor $F_*(f)\colon F_*(C) \to F_*(D)$ that on objects equals f and on hom-objects equals $F(f)$. If $\alpha\colon f \Rightarrow g$ is a V-natural transformation and $c \in \mathrm{Ob}(\mathsf{C})$, then we define $F_*(\alpha)_c$ to be the composite

$$1 \xrightarrow{\sim} F(1) \xrightarrow{F(i_a)} F(\mathsf{C}(a, b)).$$

Thus, change of base actually gives a 2-functor from the 2-category of V-categories, V-functors and V-natural transformations to the corresponding 2-category for W.

We now study how change of base affects theories and their models. We start by asking when a functor $F\colon \mathsf{V} \to \mathsf{W}$ induces a change of base $F_*\colon \mathsf{VCat} \to \mathsf{WCat}$ that "preserves enriched theories". That is, given a V-theory

$$\tau\colon \mathsf{A}_\mathsf{V} \to \mathsf{T}$$

we want to determine conditions for the base-changed functor

$$F_*(\tau)\colon F_*(\mathsf{A}_\mathsf{V}) \to F_*(\mathsf{T})$$

to induce a W-theory in a canonical way. Recall that we require V and W to be cartesian closed, equipped with chosen finite coproducts of their terminal objects. We thus expect the following conditions to be sufficient: F should be cartesian, and it should preserve the chosen finite coproducts of the terminal object:

$$F(n_\mathsf{V}) = n_\mathsf{W}$$

for all n.

Given these conditions there is a W-functor, in fact an isomorphism

$$\tilde{F}\colon \mathsf{A}_\mathsf{W} \to F_*(\mathsf{A}_\mathsf{V}).$$

On objects this maps n_W to n_V, and on hom-objects it is simply the identity from

$$\mathsf{A}_\mathsf{W}(m_\mathsf{W}, n_\mathsf{W}) = n_\mathsf{W}^{m_\mathsf{W}} = (n^m)_\mathsf{W}$$

to

$$F(\mathsf{A}_\mathsf{V}(m_\mathsf{V}, n_\mathsf{V})) = F(n_\mathsf{V}^{m_\mathsf{V}}) = F((n^m)_\mathsf{V}) = (n^m)_\mathsf{W}.$$

Using this we obtain a composite W-functor

$$\mathsf{A}_\mathsf{W} \xrightarrow{\tilde{F}} F_*(\mathsf{A}_\mathsf{V}) \xrightarrow{F_*(\tau_\mathsf{V})} F_*(\mathsf{T}).$$

This is a bijection on objects and preserves finite V-products because each of the factors has these properties. It is thus a W-theory.

Theorem 7. Let V, W be cartesian closed categories with chosen finite coproducts of their terminal objects, and let $F\colon \mathsf{V} \to \mathsf{W}$ be a cartesian functor that preserves these chosen coproducts. Then F **preserves enriched theories**: that is, for every V-theory $\tau_\mathsf{V}\colon \mathsf{A}_\mathsf{V} \to \mathsf{T}$, the W-functor

$$\tau_\mathsf{W} := F_*(\tau_\mathsf{V}) \circ \tilde{F}\colon \mathsf{A}_\mathsf{W} \to F_*(\mathsf{T})$$

is a W-theory. Moreover, F preserves models: for every model $\mu\colon \mathsf{T} \to \mathsf{C}$ of $(\mathsf{T}, \tau_\mathsf{V})$, the W-functor $F_*(\mu)\colon F_*(\mathsf{T}) \to F_*(\mathsf{C})$ is a model of $(F_*(\mathsf{T}), \tau_\mathsf{W})$.

Hence any cartesian functor that preserves chosen finite coproducts of the terminal object gives a "change of semantics" — this is a simple, ubiquitous condition, which provides for a method of translating formal languages between various "modes of operation".

7. Applications

7.1. **The SKI-combinator calculus.** The problem of substitution was noticed early in the history of mathematical foundations, even before the λ-calculus, and so Moses Schönfinkel invented **combinatory logic** [31], a basic form of logic without the red tape of variable binding, hence without functions in the usual sense. The SKI-calculus is the "variable-free" representation of the λ-calculus; λ-terms are translated via "abstraction elimination" into strings of combinators and applications. This is a technique for programming languages to minimize the subtleties of variables. A great introduction into the strange world of combinators is given by Smullyan [34].

The insight of Stay and Meredith [35] is that even though Lawvere theories have no variables, through abstraction elimination a programming language can be made into an algebraic object. When representing a computational calculus as a Gph-theory, the general rewrite rules are simply edges in the hom-graphs $t^n \to t$, with the object t serving in place of the variable. Below is the theory of the SKI-calculus:

$$\mathsf{Th(SKI)}$$

type	t	
term constructors	$S\colon$	$1 \to t$
	$K\colon$	$1 \to t$
	$I\colon$	$1 \to t$
	$(-\ -)\colon$	$t^2 \to t$
structural congruence	n/a	
rewrites	$\sigma\colon$	$(((S\ -)\ =)\ \equiv) \Rightarrow ((-\ \equiv)\ (=\equiv))$
	$\kappa\colon$	$((K\ -)\ =) \Rightarrow -$
	$\iota\colon$	$(I\ -) \Rightarrow -$

These rewrites are implicitly universally quantified; i.e. they apply to arbitrary subterms $-, =, \equiv$ without any variable binding involved, by using the cartesian structure of the category. (Here l, r denote the unitors and τ the symmetry of the product.) They are simply edges with vertices:

$$\begin{array}{lc}
(((S\ -)\ =)\ \equiv)\colon & t^3 \xrightarrow{l^{-1}\times t^3} 1\times t^3 \xrightarrow{S\times t^3} t^4 \xrightarrow{(-\ -)\times t^2} t^3 \xrightarrow{(-\ -)\times t} t^2 \xrightarrow{(-\ -)} t \\
\sigma \Downarrow & \Vert \qquad\qquad \Downarrow \qquad\qquad \Vert \\
((-\ \equiv)\ (=\equiv))\colon & t^3 \xrightarrow{t^2\times\Delta} t^4 \xrightarrow{t\times\tau\times t} t^4 \xrightarrow{(-\ -)\times(-\ -)} t^2 \xrightarrow{(-\ -)} t
\end{array}$$

$$\begin{array}{lc}
((K\ -)\ =)\colon & t^2 \xrightarrow{l^{-1}\times t^2} 1\times t^2 \xrightarrow{K\times t^2} t^3 \xrightarrow{(-\ -)\times t} t^2 \xrightarrow{(-\ -)} t \\
\kappa \Downarrow & \Vert \qquad\qquad \Downarrow \qquad\qquad \Vert \\
-\colon & t^2 \xrightarrow{t\times !} t\times 1 \xrightarrow{r} t
\end{array}$$

$$\begin{array}{lc}
(I\ -)\colon & t \xrightarrow{l^{-1}} 1\times t \xrightarrow{I\times t} t^2 \xrightarrow{(-\ -)} t \\
\iota \Downarrow & \Vert \qquad\qquad \Downarrow \qquad\qquad \Vert \\
-\colon & t \xrightarrow{t} t
\end{array}$$

A model of this theory is a power-preserving Gph-functor $\mu\colon \mathsf{Th(SKI)} \to \mathsf{Gph}$. This gives a graph $\mu(t)$ of all terms and rewrites in the SKI-calculus as follows:

$$1 \cong \mu(1) \xrightarrow{\mu(S)} \mu(t) \xleftarrow{\mu((-\ -))} \mu(t^2) \cong \mu(t)^2$$

The images of the nullary operations S, K, I are distinguished vertices of the graph $\mu(t)$, because μ preserves the terminal object which "points out" vertices. The image of the binary operation $(-\ -)$ gives for every pair of vertices $(u, v) \in \mu(t)^2$, through the isomorphism $\mu(t)^2 \cong \mu(t^2)$, a vertex $(u\ v)$ in $\mu(t)$ which is their application. In this way we get all possible terms (writing $\mu(S), \mu(K), \mu(I)$ as S, K, I for simplicity):

$$((((S\ (K\ (I\ I)))\ S)\ldots).$$

The rewrites are transferred by the enrichment of the functor: rather than functions between hom-sets, the morphism component of μ consists of graph homomorphisms between hom-graphs. So,

$$\mu_{1,t}\colon \mathsf{Th(SKI)}(1, t) \to \mathsf{Gph}(1, \mu(t))$$

maps the "syntactic" graph of all closed terms and rewrites coherently into the "semantic" graph, meaning a rewrite in the theory $a \Rightarrow b$ is sent to a rewrite in the model $\mu(a) \Rightarrow \mu(b)$.

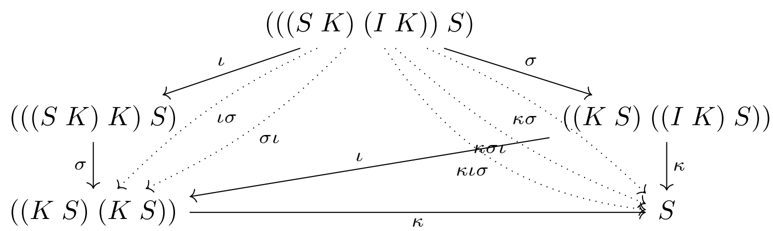
These rewrites in the image of μ are *graph transformations*, which are just like natural transformations of functors, without the commuting diagram: given two graph homomorphisms $f, g\colon G \to H$, a graph transformation $\alpha\colon f \Rightarrow g$ is a function $G_0 \to H_1$ which sends a vertex $v \in G$ to an edge $\alpha(v)$ with source $f(v)$ and target $g(v)$.

This is how μ realizes $\mathsf{Th}(\mathsf{SKI})$ as a graph of terms and rewrites: in the same way that a natural transformation of two constant functors $a \Rightarrow b\colon 1 \to \mathsf{C}$ is a morphism $a(1) \to b(1)$ in C, a rewrite of closed terms $a \Rightarrow b\colon 1 \to \mu(t)$ corresponds to an edge in $\mu(t)$:

$$\mu((I\ S))\ \bullet \xrightarrow{\mu(\iota)} \bullet\ \mu(S)$$

Finally, the fact that $\mu((-\ -))$ is not just a function but a graph homomorphism means that pairs of edges (rewrites) $(a \to b, c \to d)$ are sent to rewrites $(a\ b) \to (c\ d)$.

7.2. **Change of base.** Now we can succinctly characterize the transformation from small-step to **big-step** operational semantics. From a simple sequence of functors, we can translate between several important kinds of semantics for the SKI-calculus. For example, we have the following computation:



The solid arrows are the one-step rewrites of the initial Gph-theory; applying FC_* gives the dotted composites, and FP_* asserts that all composites between any two objects are equal. Finally, FS_* collapses the whole diagram to S. This is a simple demonstration of the basic stages of computation: small-step, big-step, full-step, and denotational semantics.

7.3. **Bisimulation.** This paper uses simple functors to illustrate the basic idea of changing semantics. Of course, there are many interesting and useful change-of-base functors. As demonstrated, any functor $F\colon \mathsf{V} \to \mathsf{W}$ which preserves finite products and finite coproducts of the terminal object can be considered as a change in semantics. For example, if we enrich in labelled directed graphs, we can utilize the important concept of *bisimulation*.

A labelled transition system consists of a set G, a label alphabet A, and a rewrite relation $\to \subset G \times A \times G$, equivalently a graph labelled by elements of A. The elements of G represent terms or processes, and the elements of A represent rewrite rules, in order to actually keep track of which kinds of rewrites are being used in a computation. An element (p, a, q) is denoted $p \xrightarrow{a} q$.

In particular, labelled transition systems allow for the correct definition of process equivalence. A **bisimilarity relation** $\equiv \subset G \times G$ consists of pairs of processes (p, q), written $p \equiv q$, defined:

$$\begin{aligned}&\forall a \in A,\ \ p', q' \in G\\ &(p \xrightarrow{a} p') \text{ implies } (\exists q' \in G\ \ (q \xrightarrow{a} q')\ \wedge\ p' \equiv q')\\ &(q \xrightarrow{a} q') \text{ implies } (\exists p' \in G\ \ (p \xrightarrow{a} p')\ \wedge\ p' \equiv q')\end{aligned}$$

Intuitively, this means that the processes p and q can always "match each other's moves" as they evolve. Then for all intents and purposes, these processes behave the same way, and hence should be considered as operationally equivalent. The **bisimulation** on G is the largest bisimilarity relation which is also a *congruence*, meaning that processes are bisimilar iff they are so in every context, i.e. when substituted into any one-hole term.

This concept, as well as the Calculus of Communicating Processes, were invented and demonstrated by Milner [26]. The latter can be expressed as an LTS-theory. The category of labelled transition systems is just like Gph, except of course we now keep track of labels. Morphisms in LTS, operations in LTS-theories, and LTS-functors all preserve labels; for example, when we compose and multiply rewrite rules, we retain this information by labelling with the actual denotation for that composite/product. Modulo these details, V = LTS is exactly like the cases considered above.

Th(CCS)

types	P		processes
	N		actions
	$\overline{N}$		coactions
operations	$0\colon$	$1 \to P$	nullity
	$\tau\colon$	$1 \to P$	internal action
	$\vert\colon$	$P^2 \to P$	parallel
	$+\colon$	$P^2 \to P$	choice
	$.\colon$	$N \times P \to P$	input
	$\bar{.}\colon$	$\overline{N} \times P \to P$	output
congruence		$(P, \vert, 0)$	commutative monoid
		$(P, +, 0)$	commutative monoid
rewrites	$tau\colon$	$. \circ (\tau \times P) \circ l^{-1} \Rightarrow id_P$	$(\tau.P \xrightarrow{tau} P)$
	$inter\colon$	$\vert \circ (. \times \bar{.}) \Rightarrow \vert$	$(a.P \vert \overline{a}.Q \xrightarrow{react} P \vert Q)$

The theory is summarized in the two rewrite rules: τ is an "unobservable" action, a process evolving in a way that is private to the ambient context; *inter* is interaction or communication - the action a is being triggered by the coaction $\overline{a}$, they are used up and the sequential processes continue in parallel. This calculus is the precursor to the π calculus [25], and is a very simple and general framework for understanding systems of interacting automata.

There is an endofunctor $B\colon \mathsf{LTS} \to \mathsf{LTS}$ which quotients by the bisimulation relation. It preserves products, $B(G \times H) \cong B(G) \times B(H)$, because $(p_1, p_2) \equiv (q_1, q_2)$ iff ($p_1 \equiv q_1$ and $p_2 \equiv q_2$). Thus we can utilize base change to perform a very useful tranformation on our semantics: from Th(CCS), we get a new theory $B_*(\mathsf{Th}(\mathsf{CCS}))$, the hom-LTS's of which consist of bisimulation equivalence classes of terms and rewrites in the calculus of communicating systems.

References

[1] J. Adámek and J. Rosický, *Locally Presentable and Accessible Categories*, Cambridge U. Press, Cambridge, 1994. (Referred to on page .)

[2] J. Baez, Can 1+1 have more than two points?, *The n-Category Café*, April 19, 2019. (Referred to on page .)

[3] J. Baez and M. Stay, Physics, topology, logic and computation: a Rosetta Stone, in *New Structures for Physics*, ed. B. Coecke, Springer, Berlin, 2011, pp. 95–172. (Referred to on page .)

[4] H.P. Barendregt, The Lambda Calculus, its syntax and semantics, in *Studies in Logic and The Foundations of Mathematics*, Elsevier, London, 1984. (Referred to on page .)

[5] M. Barr and C. Wells, *Toposes, Triples and Theories*, reprinted in *Repr. Theory Appl. Categ.* **12** (2005). (Referred to on page .)

[6] J. Beardsley and L. Z. Wong, The enriched Grothendieck construction. (Referred to on page .)

[7] C. Berger, P.-A. Melliès and M. Weber, Monads with arities and their associated theories, *J. Pure Appl. Algebra*, **216** (2012), 2029–2048. (Referred to on page .)

[8] R. Blackwell, G. M. Kelly and A. J. Power, Two-dimensional monad theory, *J. Pure Appl. Algebra* **59** (1989), 1–41. (Referred to on page .)

[9] F. Borceux, *Handbook of Categorical Algebra*, vol. 2, Cambridge U. Press, Cambridge, 1994. (Referred to on page .)

[10] R. Crole, *Categories for Types*, Cambridge U. Press, Cambridge, 1993. (Referred to on page .)

[11] B. Day and R. Street, Monoidal categories and Hopf algebroids, *Adv. Math.* **127** (1997), 99–157. (Referred to on page .)

[12] E. J. Dubuc, *Kan Extensions in Enriched Category Theory*, Lecture Notes in Mathematics, Springer, Berlin, 1970. (Referred to on page .)

[13] M. Fiore, *Abstract Syntax with Variable Binding*, Logic in Computer Science, IEEE Computer Science Press, 1999. (Referred to on page .)

[14] M. Hyland and J. Power, The category theoretic understanding of universal algebra: Lawvere theories and monads, in *Electron. Notes Theor. Comput. Sci.* **172** (2007), 437–458. (Referred to on page .)

[15] B. Jacobs, *Categorical Logic and Type Theory*, Elsevier, Amderstam, 1999. (Referred to on page .)

[16] G. M. Kelly, *Basic Concepts of Enriched Category Theory*, reprinted in *Repr. Theory Appl. Categ.* **10** (2005), 1–136. (Referred to on page .)

[17] Y. Kinoshita, J. Power and M. Takeyama, Sketches, *J. Pure Appl. Algebra* **143** (1999), 275–291. (Referred to on page .)

[18] F. W. Lawvere, Functorial semantics of algebraic theories, reprinted in *Repr. Theory Appl. Categ.* **5** (2004), 1–121. (Referred to on page .)

[19] F. E. J. Linton, Some aspects of equational theories, in *Proceedings of the Conference on Categorical Algebra*, eds. S. Eilenberg et al., Springer, Berlin, 1965. (Referred to on page .)

[20] R. B. B. Lucyshyn-Wright, Enriched algebraic theories and monads for a system of arities, *Theory Appl. Categ.* **31** (2016), 101–137. (Referred to on page .)

[21] C. Lüth and N. Ghani, Monads and modular term rewriting, *Category Theory and Computer Science (Santa Margherita Ligure, 1997)*, Springer, Berlin, 1997, pp. 69–86. (Referred to on page .)

[22] S. Mac Lane, *Categories for the Working Mathematician*, Springer, Berlin, 1998. (Referred to on page .)

[23] L. G. Meredith and M. Radestock, A reflective higher-order calculus, *Electronic Notes in Theoretical Computer Science* **141** (2005), 49–67. (Referred to on page .)

[24] B. Milewski, *Category Theory for Programmers*, Chap. 14: Lawvere theories, 2017. (Referred to on page .)

[25] R. Milner, The polyadic π-calculus: a tutorial, in *Logic and Algebra of Specification*, Springer, Berlin, 1993, 203–246. (Referred to on page .)

[26] R. Milner, Communicating and Mobile Systems: The Pi Calculus, in *Cambridge University Press*, Cambridge, UK, 1999. (Referred to on page .)

[27] K. Nishizawa and J. Power, Lawvere theories enriched over a general base, *J. Pure Appl. Algebra* **213** (2009), 377–386. (Referred to on page .)

[28] G. D. Plotkin, A structural approach to operational semantics, *J. Log. Algebr Program.* **60/61** (2004) 17–139. (Referred to on page .)

[29] M. Hyland and J. Power, Discrete Lawvere theories and computational effects, in *Theoretical Comp. Sci.* **366** (2006), 144–162. (Referred to on page .)

[30] J. Power, Enriched Lawvere theories, *Theory Appl. Categ.* **6**(1999), 83–93. (Referred to on page .)

[31] M. Schönfinkel, Über die Bausteine der mathematischen Logik, *Mathematische Annalen* **92** (1924), 305–316. (Referred to on page .)

[32] R. A. G. Seely, Modelling computations: a 2-categorical framework, in *Proceedings of the Second Annual IEEE Symposium on Logic in Computer Science (LICS 1987)*, IEEE Computer Society Press, Ithaca, New York, pp. 22–25. (Referred to on page .)

[33] P. Selinger, Lecture notes on the lambda calculus. (Referred to on page .)

[34] R. Smullyan, *To Mock a Mockingbird: and Other Logic Puzzles Including an Amazing Adventure in Combinatory Logic*, Oxford U. Press, Oxford, 2000. (Referred to on page .)

[35] M. Stay and L. G. Meredith, Representing operational semantics with enriched Lawvere theories. (Referred to on page .)

[36] M. Stay and L. G. Meredith, Logic as a distributive law. (Referred to on page .)