Powermonads and Tensors of Unranked Effects

Sergey Goncharov and Lutz Schröder
DFKI Bremen, Germany
Email: Sergey.Goncharov@dfki.de, Lutz.Schroeder@dfki.de

Abstract
In semantics and in programming practice, algebraic concepts such as monads or, essentially equivalently, (large) Lawvere theories are a well-established tool for modelling generic side-effects. An important issue in this context are combination mechanisms for such algebraic effects, which allow for the modular design of programming languages and verification logics. The most basic combination operators are sum and tensor: while the sum of effects is just their non-interacting union, the tensor imposes commutation of effects. However, for effects with unbounded arity, such as continuations or unbounded nondeterminism, it is not a priori clear whether these combinations actually exist in all cases. Here, we introduce the class of uniform effects, which includes unbounded nondeterminism and continuations, and prove that the tensor always exists if one of the component effects is uniform, thus in particular improving on previous results on tensoring with continuations. We then treat the case of nondeterminism in more detail, and give an order-theoretic characterization of effects for which tensoring with nondeterminism is conservative, thus enabling nondeterministic arguments such as a generic version of the Fischer-Ladner encoding of control operators.

1. Introduction
Both in actual programming languages and in their semantics and meta-theory, one encounters a wide variety of phenomena that can be subsumed under a broadly understood notion of side-effect, such as various forms of state, input/output, resumptions, backtracking, nondeterminism, continuations, and many more. This proliferation of effects motivates the search for generic frameworks that encapsulate the exact nature of side-effects and support abstract formulations of programs (such as Haskell’s generic while-loop), semantic principles, and program logics. A fairly well-established abstraction of this kind is the modelling of side-effects as monads, following seminal work by Moggi [16]; this principle is widely used in programming language semantics (e.g. [8], [13], [18], [23]) and moreover underlies the incorporation of side-effects in the functional programming language Haskell [27]. Besides supporting generic results that can be instantiated to particular effects at little or no cost, monads allow for a clear delineation of the scope of effects [6], [17]. A more recent development is the advancement of Lawvere theories [14] for the generic modelling of effects, thus emphasizing their algebraic nature [21].

One of the advantages of these approaches is that they provide for a modular semantics of effects. It has been observed that many effects, such as state, exceptions, and continuations, induce so-called monad transformers that can be seen as adding the respective effect to a given set of effects [2], [15]; again, the notion of monad transformer plays a central role in Haskell. More recently, it has been shown that many monad transformers arise from binary combination operators that join effects in a prescribed way. The most important among these constructions are the sum of effects, which corresponds simply to the disjoint union of algebraic theories, and the tensor, which additionally imposes a commutation condition [9], [11]. E.g., the exception monad transformer is summation with the exception monad, and the state monad transformer is tensoring with the state monad. These combination methods are often mixed; e.g. [24] uses both sums and tensors of nondeterminism with other effects.

One of the problems that arise with sum and tensor in the context of large Lawvere theories, i.e. theories that can be unranked in the sense that their operations have unbounded arities, such as unbounded nondeterminism or continuations, is that it is not clear that the combined theories actually exist in all cases. Specifically, it is known that sums of unranked theories need not exist [12], and is has been conjectured that the tensor of two large Lawvere theories need not exist, even when one of them is in fact ranked [9]. In the present work, we introduce the notion of uniform
theory, and prove that the tensor of two large Lawvere theories always exists if one of them is uniform. The class of uniform theories includes several variants of nondeterminism (e.g. unbounded and countable, but not finite) as well as, somewhat surprisingly, continuations; thus, our existence result improves a previous result stating that the tensor of any ranked theory with continuations always exists [9].

One may read this result as yielding a number of new monad transformers; we are particularly interested in the nondeterminism monad transformer, which we dub the powermonad. This leads us to a second problem associated specifically with the tensor: since the tensor imposes complex algebraic interaction between the component effects, it cannot in general be expected to be conservative in the sense that the maps from the components into the tensor are monic.

To deal with this issue in the special case of nondeterminism, we focus on pointed theories L, which come with a natural approximation ordering. We begin by giving a simplified construction of tensoring with nondeterminism, in which morphisms in the tensor of L with nondeterminism are sets of morphisms modulo rectangular equivalence, a comparatively simple equivalence that forces uniqueness of tupling morphisms. From there, we obtain a more order-theoretic description of the tensor in terms of closed sets of L-morphisms, which leads to a simple characterization of theories for which tensoring with nondeterminism is order-theoretically conservative.

The main reason for our interest in tensoring with nondeterminism is that it yields exactly the free extension of a given theory to a completely additive theory, i.e. one that is enriched over complete join semilattices; in terms of nondeterministic programming, being additive amounts to having nondeterministic choice operators that distribute over sequential composition (on both sides, hence providing a trace-based rather than a bisimulation-based perspective). Thus, whenever a theory L can be conservatively tensored with nondeterminism, one can conduct equational and order-theoretic proofs in it pretending that L is completely additive.

E.g., one can use the well-known Fischer-Ladner encoding

\[
\text{if } b \text{ then } p \text{ else } q := b?; p + (\neg b)?; q \\
\text{while } b \text{ do } p := (b?; p); (\neg b)\
\]

(which translates imperative constructs into a rather simple algebraic framework, and moreover makes their nondeterministic flavour [20] explicit) generically, i.e. for any effect satisfying our conservativity conditions.

Related Work

The use of Lawvere theories in general and the tensor operation in particular for modelling computational effects has been promoted in [10], [21]. Tensoring with continuations is the main theme of [9], where it is proved that the tensor of a ranked theory (i.e. with operations of bounded arity) with the theory of continuations always exists. Moreover, it is shown in [11] that the tensor of global state with any other (large) Lawvere theory always exists, and corresponds to the standard monad transformer for global state. It has been conjectured that the tensor of two large Lawvere theories may fail to exist even one of them is ranked [9, p. 30]. Previous work on the specific combination of unbounded nondeterminism and probabilistic choice uses a different form of interaction than imposed by the tensor [25], [26].

2. Large Lawvere Theories and Monads

In a nutshell, the principle of monadic encapsulation of side-effects originally due to Moggi [16] and subsequently introduced into the functional programming language Haskell as the principal means of dealing with impure features [27] consists in moving the side effect from the function arrow into the result type of a function: a side-effecting function \(X \rightarrow Y\) becomes a pure function \(X \rightarrow TY\), where \(TY\) is a type of side-effecting computations over \(Y\); the base example is \(TY = S \rightarrow (S \times Y)\) for a fixed set \(S\) of states, so that functions \(X \rightarrow TY\) are functions that may read and update a global state (more examples will be given later). Formally, a monad on the category of sets, presented as a Kleisli triple \(\mathbb{T} = (T, \eta, \Delta^*)\), consists of a function \(T\) mapping sets \(X\) (of values) to sets \(TX\) (of computations), a family of functions \(\eta_X : X \rightarrow TX\), and a map assigning to every function \(f : X \rightarrow TY\) a function \(f^* : TX \rightarrow TY\) that lifts \(f\) from \(X\) to computations over \(X\). These data are subject to the equations \(\eta^* = \text{id}, f^* \eta = f, (f^* g)^* = f^* g^*\), which ensure that the Kleisli category of \(\mathbb{T}\), which has sets as objects and maps \(X \rightarrow TY\) as morphisms, is actually a category, with identities \(\eta : X \rightarrow TY\) and composition \(f \circ g\). On Set, all monads are strong, i.e. equipped with a natural transformation \(X \times TY \rightarrow T(X \times Y)\) satisfying a number of coherence conditions [16].

Monads were originally intended as abstract presentations of algebraic theories, with \(TX\) abstracting the free algebra over \(X\), i.e. terms over \(X\) modulo provable equality. It has been shown that the algebraic view of monads gives rise to computationally natural operations for effects; e.g. the state monad (with state
set $S = V^L$ for sets $V$ of values and $L$ of locations) can be algebraically presented in terms of operations \textit{lookup} and \textit{update} \cite{21}. Categorically, this shift of viewpoint amounts to generating monads from Lawvere theories. To cover unranked theories, we use the notion of large Lawvere theory \cite{4}, introduced into the theory of generic effects in \cite{9}. Generally, we denote hom-sets of a category $C$ in the form $C(A, B)$.

\textbf{Definition 2.1} (Lawvere theory). A \textit{large Lawvere theory} is given by a locally small category $C$ with small products, together with a strict product preserving identity-on-objects functor $I : \text{Set}^{\text{op}} \rightarrow L$. We call $I$ the \textit{indexing functor}, and we denote $I f$ by $[f]$ for a map $f$. A morphism of large Lawvere theories $L_1 \rightarrow L_2$ is a functor $L_1 \rightarrow L_2$ that commutes with the indexing functors (and hence preserves small products). A \textit{model} of a large (finitary) Lawvere theory $L$ in a category $C$ with small products is a small product preserving functor $L \rightarrow C$.

The algebraic intuition behind these definitions is that the objects of a Lawvere theory are sets of variables (typically denoted $n, m, k, \ldots$), and morphisms $n \rightarrow m$ are $m$-tuples of terms over $n$, or substitutions from $m$ into terms over $n$. The indexing functor prescribes the effect of rearranging variables in terms. The notion of model recalled above implies that Lawvere theories provide a representation of effects that is independent of the base category $C$, and given enough structure on $C$ a Lawvere theory will induce a monad on $C$. E.g., in categories of domains, the theory of non-blocking nondeterminism (Example 2.2.2 below) induces precisely the Plotkin powerdomain monad (while the Hoare and Smyth powerdomains require enriched Lawvere theories) \cite{1}.

It is well-known that large Lawvere theories and strong monads on $\text{Set}$ form equivalent (overlarge) categories \cite{4}, \cite{9}. The equivalence maps a large Lawvere theory $L$ to the monad $T_L X = L(X, 1)$ (we elide the full description), and a monad $T$ to the dual of its Kleisli category. We therefore largely drop the distinction between monads and Lawvere theories, and freely transfer concepts and examples from one setting to the other; occasionally we leave the choice open by just using the term \textit{effect}.

We say that a Lawvere theory $L$ is \textit{ranked} if it can be presented by operations (and equations) of arity less than $\aleph$ for some cardinal $\aleph$; otherwise, $L$ is \textit{unranked}. Categorically, $L$ having rank $\aleph$ amounts to preservation of $\aleph$-directed colimits by the induced monad.

\textbf{Example 2.2}. 1) \textit{Global state}: as stated initially, $T X = S \rightarrow (S \times X)$ is a monad (for this and other standard examples, we omit the description of the remaining data), the well-known \textit{state monad}. A variant is the \textit{partial state monad} $T X = S \rightarrow (S \times X)_\bot$, where $X_\bot$ extends $X$ by a fresh element $\bot$ representing non-termination. (This induces a relational model of non-termination in the spirit of PDL and related formalisms; a domain-theoretic treatment of non-termination requires a domain-enriched Lawvere theory in which $\bot$ is explicitly a bottom element).

2) \textit{Nondeterminism}: the \textit{unranked} large Lawvere theory $L_{\text{np}}$ for nondeterminism arises from the powerset monad $\mathcal{P}$. It has $m$-tuples of subsets of $n$ as morphisms $n \rightarrow m$. Variants arise on the one hand by restricting to nonempty subsets, thus ruling out non-termination, and on the other hand by bounding the cardinality of subsets. We denote nonemptiness by a superscript $\ast$, and cardinality bounds by subscripts. E.g., the large Lawvere theory $L_{\text{np}}^{\ast \kappa}$ describes finite non-blocking nondeterminism; its morphisms $n \rightarrow m$ are $m$-tuples of nonempty finite subsets of $n$. Yet another variant arises by replacing sets with multisets, i.e. maps $X \rightarrow [\aleph \cup \{\infty\}]$, thus modelling weighted nondeterminism \cite{3} as a Lawvere theory $L_{\text{wnt}}$.

3) \textit{Continuations}: The continuation monad maps a set $X$ to the set $(X \rightarrow R) \rightarrow R$, for a fixed set $R$ of results. The corresponding \textit{unranked} large Lawvere theory $L_{\text{cont}}^{\ast \kappa}$ has maps $m \rightarrow ((n \rightarrow R) \rightarrow R)$ as morphisms $n \rightarrow m$.

4) \textit{Input/Output}: For a given set $I$ of input symbols, the Lawvere theory $L_I$ for input is generated by a single $I$-ary operation; it is an \textit{absolutely free} theory, i.e. has no equations. Similarly, given a set $O$ of output symbols, the Lawvere theory $L_O$ for output is generated by unary operations $f_o$ for $o \in O$.

Further effects that fit the algebraic framework are exceptions ($T X = X + E$), resumptions ($R X = \mu Y. T(X + Y)$ for a given base effect $T$) and many more.

\textbf{Notation 2.3}. Let $L$ be a large Lawvere theory. For an object $n$ of $L$ and $i \in n$, we let $\varepsilon_i$ denote the map $1 \rightarrow n$ that picks $i$. Thus, the $\varepsilon_i$ induce product projections $[\varepsilon_i] : n \rightarrow 1$ in $L$. Given two sets $n$ and $m$, their $\text{Set}$-product $n \times m$ can be viewed as the sum of $m$ copies of $n$ in $\text{Set}$, and hence as the $m$-th power of $n$ in $L$. This induces for every $f : n \rightarrow k$ in $L$ the morphisms $f \otimes m : n \times m \rightarrow k \times m$ and $m \otimes f : m \times n \rightarrow m \times k$.

A convenient way of denoting generic computations is the so-called \textit{computational metalanguage} \cite{16}, which has found its way into functional programming in the shape of Haskell’s do-notation. We briefly outline the
version of the metalanguage we use below.

The metalanguage serves to denote morphisms in the underlying category of a given monad, using the monadic structure; since large Lawvere theories correspond to monads on Set, the metalanguage just denotes maps in our setting. We let a signature \( \Sigma \) consist of a set \( B \) of base types, to be interpreted as sets, and a collection of typed function symbols to be interpreted as functions. Here, we assume that types \( A, B \in T \) are generated from the base types by the grammar

\[
A, B ::= 1 \mid A \in B \mid A + B \mid A \times B \mid TA
\]

where + and \( \times \) are interpreted as set theoretic sum and product, respectively, \( 1 \) is a singleton set, and \( T \) is application of the given monad. We then have standard formation rules for terms-in-context \( \Gamma \vdash t : A \), read ‘term \( t \) has type \( A \) in context \( \Gamma \)’, where a context is a list \( \Gamma = (x_1 : A_1, \ldots, x_n : A_n) \) of typed variables (later, contexts will mostly be omitted):

\[
x : A \in \Gamma \quad f : A \to B \in \Sigma \quad \Gamma \vdash t : A
\]

\[
\Gamma \vdash x : A \\
\Gamma \vdash f(t) : B \\
\Gamma \vdash \star : 1
\]

\[
\Gamma \vdash t : A \\
\Gamma \vdash u : B \\
\Gamma \vdash (t, u) : A \times B
\]

\[
\Gamma \vdash \text{fst}(t) : B \\
\Gamma \vdash \text{snd}(t) : B
\]

\[
\Gamma \vdash s : A + B \\
\Gamma \vdash y : B \vdash u : C
\]

\[
\Gamma \vdash \text{case } s \text{ of } \text{inl} x \mapsto t; \text{inr } y \mapsto u : C
\]

\[
\Gamma \vdash t : A \\
\Gamma \vdash \text{inl } t : A + B
\]

This syntax supports, e.g., the standard encoding of the if-operator as

\[
\text{if } b \text{ then } p \text{ else } q = \text{case } b \text{ of } \text{inl } s \mapsto p; \text{inr } q \mapsto q,
\]

for \( b : 2 \), where \( 2 = 1 + 1 \). Beyond this, we have monadic term constructors

\[
\Gamma \vdash \text{ret } t : TA \\
\Gamma \vdash \text{do } x \leftarrow p ; q : TB
\]

called return and binding, respectively. Return is interpreted by the unit \( \eta \) of the monad, and can be thought of as returning a value. A binding \( \text{do } x \leftarrow p ; q \) executes \( p \), binds its result to \( x \), and then executes \( q \), which may use \( x \) (if not, mention of \( x \) may be omitted). It is interpreted using Kleisli composition and strength, where the latter serves to propagate the context \( \Gamma \) [16]. In consequence, one has the monad laws

\[
\text{do } x \leftarrow p ; \text{ret } x = p \\
\text{do } x \leftarrow \text{ret } a ; p = p[a/x]
\]

\[
\text{do } x \leftarrow (\text{do } y \leftarrow p ; q) ; r = \text{do } x \leftarrow p ; y \leftarrow q ; r
\]

Terms of a type \( TA \) are called programs; a ground program is a program in the empty context.

### 3. Tensors of Large Lawvere Theories

One of the key benefits of the monadic modelling of effects is that it allows for a modular treatment, where effects are combined from basic building blocks according to the demands of the programming task at hand. In current programming practice (specifically in Haskell [19]), this is typically achieved by generalizing a given effect to a monad transformer [15], [16], i.e. a function that maps monads to monads, in the process extending them with a given effect. For instance, the state monad transformer \( ST \) for a given set \( S \) of states maps a given monad \( T \) to the monad \( ST(T) \) with \( ST(T)(x) = S \to T(S \times X) \). Monad transformers are very general, but do not support a great deal of meta-theoretic results, as no further properties are imposed on them; e.g., they need not be functorial. It has been shown in [11] that many monad transformers arise from a few basic binary operations on Lawvere theories (equivalently on monads). E.g., the exception monad transformer, which maps a monad \( T \) to the monad \( T(\star + E) \) for a fixed set \( E \) of exceptions, is just summation with \( \star + E \); expressed in terms of large Lawvere theories, the sum \( L_1 + L_2 \) of two effects \( L_1, L_2 \) is simply the disjoint union of the associated theories, i.e. universal w.r.t. having morphisms \( L_1 \to L_1 + L_2 \leftarrow L_2 \). Another important operation is the tensor which additionally imposes a strong form of interaction between the component theories in the form of a commutation law.

**Definition 3.1** (Tensor product). [9] Given Lawvere theories \( L_1 \) and \( L_2 \), their tensor product \( L_1 \otimes L_2 \) is the Lawvere theory possessing the universal property of having commuting morphisms of Lawvere theories \( L_1 \to L_2 \otimes L \leftarrow L_2 \) (elided in the notation): given \( f_1 : n_1 \to m_1 \) in \( L_1 \) and \( f_2 : n_2 \to m_2 \) in \( L_2 \), we demand satisfaction of the tensor law, i.e. commutativity of the diagram

\[
\begin{array}{ccc}
T_n & \xrightarrow{f_1 \otimes f_2} & T_{m_1} \\
\downarrow n_1 \times n_2 & & \downarrow m_1 \times m_2 \\
T_{n_1} \times T_{n_2} & \xrightarrow{1 \times T} & T_{n_1 \times n_2} \\
\end{array}
\]

By the equivalence between large Lawvere theories and monads, this induces also a notion of tensor of monads [9]. The computational meaning of the commutation condition becomes clearer in the computational metalanguage: if we extended the metalanguage with subtypes \( T_nA \) of \( TA \) interpreted using the component monads \( T_1, T_2 \) of the tensor \( T = T_1 \otimes T_2 \), it amounts
to the equality

$$\text{do } x_1 \leftarrow p_1; x_2 \leftarrow p_2; \text{ret}(x_1, x_2) =$$
$$\text{do } x_2 \leftarrow p_2; x_1 \leftarrow p_1; \text{ret}(x_1, x_2)$$

in context $\Gamma_1, \Gamma_2$, where $\Gamma_i \triangleright p_i : T_i A_i$ for $i = 1, 2$; i.e. programs having only one of the component effects do not interfere with programs having only the other effect.

**Example 3.2.** [11] Tensoring with the state monad $TX = S \to (S \times X)$ yields exactly the standard state monad transformer.

For large Lawvere theories, it is not in general clear that sum and tensor exist. E.g., the sum of the theory generated by a single unary operation and no equations with almost any unranked Lawvere theory fails to exist [12]. Existence of sum and tensor is a size issue – if arities of operations are unbounded, then the terms over a given set of variables need not form a set. Generally, the tensor has a better chance to exist than the sum, since it introduces additional equations [9]. Nevertheless, it has been conjectured that even when one of the component theories is ranked, the tensor does not exist in general (of course, it does exist in case both components are ranked). We proceed to show that the tensor exists whenever one of the component theories is uniform in the sense defined presently.

**Definition 3.3 (Uniformity).** Let $L$ be a large Lawvere theory. The constants of $L$ are the elements of $c_L := L(0, 1)$. For every set $n$ we denote by $c^n_L : n \to n + c_L$ the morphism $[\text{id}] \times \prod_{j \in c_L} f$. We say that $L$ is uniform if for every $L$-morphism $f : n \to m$ there exists a generic morphism, i.e. a morphism $\hat{f} : k \to 1$ for some set $k$ such that there exists a set-function $u : k \times m \to n + c_L$ with $f = (f \otimes m) \circ [u] \circ c^n_L$.

In other words, a theory is uniform if all terms over a given set of variables can be obtained from a single generic term $\hat{f}$, possibly having more variables, by substituting for the variables of $\hat{f}$ either variables from $n$ or constants. The relevance to existence of tensors is clear: if a theory $L_2$ is uniform, then the tensor law of a putative tensor $L_1 \otimes L_2$ can always be made to apply to a term that has, say, a top layer of operations from $L_1$ whose arguments have a top layer from $L_2$.

**Remark 3.4.** It is easy to see that in Definition 3.3, $k$ can be bounded by $(n + c)^m$.

**Example 3.5.**
1) The theory $L_{P^*}$ of non-blocking unbounded nondeterminism is uniform: Recall that a morphism $f : n \to m$ in $L_{P^*}$ is a family of $m$ nonempty subsets of $n$. As a generic morphism $\hat{f}$ for $f$, we can thus take the full set $n$, seen as a morphism $n \to 1$, from which any other subset of $n$ can be obtained by identifying some of the variables.

2) The theory $L_P$ of unbounded nondeterminism is uniform: The argument is analogous as for $L_{P^*}$, except that we now need to use also the constant $\varnothing$ in substitutions in order to obtain the empty set as a substitution instance of the generic morphism $\hat{f}$.

3) The theory $L_{P_n}$ of finite nondeterminism fails to be uniform: if $\sup_{i \in [m]} A_i = \infty$ for an infinite family $(A_i)_{i \in [m]}$ of finite sets of an infinite set $n$, then there is no single finite set from which all sets $A_i$ can be obtained by substituting variables from $n$ or $\varnothing$.

4) The theory $L_{P_{\omega_1}}$ of countable nondeterminism is uniform: any infinite countable subset of $n$ will serve as a generic morphism $\hat{f}$ for any morphism $f : n \to m$, i.e. any family of at $m$ most countable subsets of $n$.

5) The theory $L_{mult}$ of unbounded weighted nondeterminism is uniform: Recall from Example 2.2.2 that a morphism $f : n \to m$ in $L_{mult}$ is a family of multisets over $n$ (i.e. maps $n \to \mathbb{N} \cup \{\infty\}$). As a generic morphism $\hat{f}$, we can take the multiset over $\mathbb{N} \times n$ that contains every element with multiplicity 1.

Moreover, uniformity also subsumes continuations, a fact that we state and prove separately.

**Lemma 3.6.** For every $R$, the continuation theory $L_{cont}^R$ (Example 2.2.3) is uniform.

**Proof:** W.l.o.g. assume $|R| \geq 2$. We identify the set of constants of $L_{cont}^R$ with $R$. Let $f : n \to m$ in $L_{cont}^R$, recall that $L_{cont}^R$ is the dual of the Kleisli category of the continuations monad, i.e. $f$ is a map $m \to ((n \to R) \to R)$. Pick $J$ such that $|n| \leq |R^J|$; w.l.o.g. $m = R^J$. The required generic morphism for $f$ is $\hat{f} : n + J \to 1$, defined by

$$\hat{f}(c) = f(\lambda j. c(\text{inr } j)) (\lambda a. c(\text{inl } a))$$

for $c : n + J \to R$: Let $u : (n + J) \times m \to n + R$,

$$u(x, i) = \begin{cases} \text{case } x \text{ of } \text{inl } y \mapsto \text{inl } y; \text{inr } j \mapsto \text{inr } i(j). \end{cases}$$

Then for $i \in m = R^J$ and $k : n \to R$,

$$(\hat{f} \otimes m)[u]c^n_L(i)(k) = f(\lambda x. \text{case } u(x, i) \text{ of } \text{inl } y \mapsto k(y); \text{inr } r \mapsto r)$$
$$= f(\lambda x. \text{case } x \text{ of } \text{inl } y \mapsto k(y); \text{inr } j \mapsto i(j))$$
$$= f(i)(k). \quad \square$$

The main existence result for tensors is as follows.

**Theorem 3.7.** Let $L_1$ and $L_2$ be two Lawvere theories, and let $L_2$ be uniform. Then the tensor product $L_1 \otimes L_2$ exists.
Proof sketch: By explicit syntactic construction of the tensor product $L = L_1 \otimes L_2$. One constructs a precursor $C$ of the tensor with its morphisms $n \to m$ are equivalence classes of paths $n \to m$; a single step $k \to l$ in a path is of the form $f \star g$ where $f : p \to l$ in $L_2$ and $g : k \to p$ in $L_1$. The equivalence is the congruence $\sim$ on paths $(f_1 \star g_1, \ldots, f_n \star g_n)$ generated by $\langle [id] \star [id] \rangle \sim \langle 0 \rangle, (f \star [e] \star g) \sim (f \star [c] \star g)$, and $(f \star (n' \otimes g))(f' \otimes ((m') \star g')) \sim (f \star f' \otimes m') \star ((n \otimes g) g')$.

Using uniformity of $L_2$, one shows that every morphism of $C$ has a representative of the form $(f \star g \mid \exists! \sigma \in \mathbb{C})$ (1)

(recall notation from Definition 3.3). One shows moreover that in (1), the domain of $f$ can be taken to be $k = L_1(n + c_{L_2}, 1)$, so that $C$ is locally small. One defines a functor $I : \text{Set}^{op} \to C$ by $I = [e] \star [id]$. It turns out that $I$ maps products to weak products, i.e. factorizations through the product structure but need not be unique; this is amended by further quotienting. \(\square\)

**Corollary 3.8.** For a large Lawvere theory $L_1$, the tensor $L_1 \otimes L_2$ exists if $L_2$ is one of the following theories:

- unbounded or countable nondeterminism $L_{P, \infty}$; $L_{P, \infty}$;
- unbounded or countable non-blocking nondeterminism $L_{P, \infty}$; $L_{P, \infty}$;
- weighted nondeterminism $L_{\text{mult}}$; or
- continuations $L_{\text{cont}}$.

Of course, a corresponding result holds for monads. This result induces new monad transformers for nondeterminism, continuations, etc. The existence result for tensoring with continuations improves over previous results stating that the tensor of continuations with any ranked theory with exist [9]. The results involving nondeterminism are, to our knowledge, entirely new. We refer to tensoring with nondeterminism as the powermonad construction.

4. Completely Additive Monads and the Fischer-Ladner Encoding

Having shown that the tensor of any effect with nondeterminism always exists, we proceed to show that this amounts to a universal construction of an additive theory, i.e. a theory that includes nondeterministic choice operators which distribute over sequential composition. There are two versions of this phenomenon, with and without blocking (i.e. the empty set); for economy of presentation we concentrate on the case with blocking. We start out with a few notions concerned with blocking.

**Definition 4.1** (Pointed theory). We call a large Lawvere theory $L$ pointed if $|L(0, 1)| = 1$.

Pointedness is connected to tensoring, as follows.

**Definition 4.2.** We denote by $L_\perp$, the large Lawvere theory generated by a constant $\perp$ and no equations.

**Lemma 4.3.** A large Lawvere theory $L$ is pointed iff $L \otimes L_\perp \cong L$.

We denote the only constant of a pointed theory by $\perp_{0,1}$, and put $\perp_{n,m} = (\perp_{0,1} \otimes m)[\perp] : n \to m$ for all $n, m$, where $\perp$ is the unique map $0 \to n$. In the sequel, we mostly write $\perp$ in place of $\perp_{n,m}$. As usual, we have a corresponding notion of pointed monad.

**Example 4.4** (Pointed effects). Besides $L_\perp$, basic examples of pointed theories include all forms of non-determinism with blocking. Similarly, the list monad is pointed ($\perp$ is the empty list). By Lemma 4.3, the state monad transformer, being defined by tensoring, preserves pointedness; e.g. the monads $S \to ((S \times \perp) + 1)$ and $S \to P(S \times \perp)$ are pointed.

The theories $L_1$, $L_\perp$ for input and output, respectively, have no constants and hence are not pointed. However, one can easily form pointed variants by tensoring with $L_\perp$. Since $\perp$ is strict in the tensor, this amounts to imposing rollback in the case of blocking.

The basic notion of a theory with nondeterminism is now as follows.

**Definition 4.5** (Additive Lawvere theories). [7] A Lawvere theory $L$ is finitely additive if $L$ is enriched over join semilattices, and completely additive if $L$ is enriched over complete join semilattices (with $\perp$).

Again, corresponding notions for monads are implied. Joins serve to model nondeterministic choice; enrichment includes the condition that choice and deadlock distribute over composition. Note that this corresponds to the form of nondeterminism appearing in nondeterministic Turing machines: executions that either block or fail to terminate are discarded entirely. The relation to tensors is the following.

**Lemma 4.6.** For a large Lawvere theory $L$, the following are equivalent.

i) $L$ is completely additive.

ii) $L \cong L \otimes L_{\perp}$.

iii) For every $n$, $L(n, 1)$ contains a distinguished morphism $U_n : n \to 1$ such that for any surjection $\sigma : m \to n$,

$$U_n = U_m[\sigma].$$
and for every L-morphism \( f : m \to 1 \),
\[ f \circ (U_n \otimes m) = U_n \circ (n \otimes f). \]

Moreover, \( L \) is pointed, and \( U_0 = \bot \).
The same equivalence holds for finite additivity, tensoring with \( L^\infty \), and (iii) for finite \( n, m \).

The operations \( U_n \) are \( n \)-fold joins, with 0-fold join being \( \bot \). In other words, a completely additive Lawvere theory \( L \) is one that has nondeterministic choice operators that commute over all operations of \( L \) as prescribed by the tensor law. From the above, it is immediate that
tensoring a large Lawvere theory \( L \) with \( L_P \) yields the free completely additive Lawvere theory over \( L \),

i.e. the (overlarge) category of completely additive Lawvere theories is reflexive in the category of Lawvere theories.

**Example 4.7.** By the above, the generic example of a completely additive monad is \( P \), with joins being set unions. Another example is the combination of nondeterminism and global state, \( S \to P(S \times \_\_) \).

**Remark 4.8.** Although completely additive theories are enriched, they can be treated as standard large Lawvere theories – as made explicit in Lemma 4.6, the completely additive structure is algebraic (although unranked), and hence respected by all product-preserving functors.

We proceed to formalize the example application from the introduction, i.e. to show that completely additive monads indeed allow for a generic Fischer-Ladner encoding of control structures. We base this formalization on the fact that every completely additive monad is a Kleene monad [7], i.e. supports Kleene iteration. Specifically, we can extend the computational metalanguage with operators \( \bot \) (deadlock), + (binary choice, interpreted by binary joins), and a generic loop construct \( \text{init}\_x \leftarrow p \text{in} q^* \) (in which \( x \) is a bound variable). The latter is interpreted as the join of all finite iterations of \( q \), prefixed with \( x \leftarrow p \) and with the result \( x \) of the computation fed through the loop; i.e. \( \text{init}\_x \leftarrow p \text{in} q^* \) is the join of \( p \) and all programs \( x \leftarrow p; x \leftarrow q; \ldots; x \leftarrow q; q \) where \( x \leftarrow q \) appears \( n \geq 0 \) times. Of course, + and \( \bot \) are supported already by finitely additive monads. A sound axiomatization of this extended language [7], extending the basic axiomatization of the computational metalanguage and separated into axioms for nondeterminism and Kleene iteration, is given in Fig. 1. The ordering \( \leq \) is, as usual, defined via \( p \leq q \iff p+q=q \). Choice and Kleene iteration allow for defining \( \text{if} \) and \( \text{while} \) primitives over \( P \otimes T \) in the Fischer-Ladner style, as announced in the introduction. We include in the signature the test operator \( ? : 2 \to T \), which sends \( \text{in} \_\* \) to \( \text{ret} \_\* \) and \( \text{inl} \_\* \) to \( \bot \). Recall that we have given a definition of the if-operator in terms of the case operator in the base language (p. 4), with \( 2 = 1 + 1 \) representing the Booleans.

**Proposition 4.9** (Generic Fischer-Ladner encoding).

1) Let \( T \) be a finitely additive monad. Then for \( b : 2 \) and \( p, q : TA \),
\[ \text{if } b \text{ then } p \text{ else } q = \text{do } b?; p + \text{do } (\neg b)?; q. \]

2) Let \( T \) be a Kleene monad and let \( \Gamma, x : A \rhd b : 2 \), \( \Gamma, x : A \rhd p : TA \) be such that the map \( \Gamma, x : A \rhd q : TA \mapsto if b then do x \leftarrow p; q \text{ else ret } x \)
has a least fixed point \( \text{while}(b, p) \text{ w.r.t. } \leq \). Then
\[ \text{while}(b, p) = \text{do } x \leftarrow \text{init}\_x \leftarrow \text{ret} x \text{in}(\text{do } b?; p^*) \}; (\neg b)?; \text{ret} x. \]

**5. Nondeterminism, Conservatively**

As indicated above, the algebraic complexity of the tensor implies that it is, in general, not at all clear that the component theories inject into the tensor, i.e. that adding a new effect is conservative. In the following we investigate this issue for the case of nondeterminism, i.e. for the powerMonad; in the terminology of the previous section, this amounts to asking for which monads \( T \) it can be soundly assumed that they are additive, thus enabling, e.g., arguments using the Fischer-Ladner encoding.

Since \( L_P \) has a constant (\( \emptyset \)), one obvious case where conservativity fails is if \( T \) has more than one constant. However, this is not the only problem: Even tensoring with nonempty powerset \( P^* \) can be non-conservative, one counterexample being \( (P^*)^2 \otimes P^* = P^* \) where \( (P^*)^2 = P^*P^* \) is the double nonempty powerset monad (which may be thought of as generated by unbounded conjunction and disjunction operators and a distributive law). Collapse of \( (P^*)^2 \otimes P^* \) to \( P^* \) is due to a variant of the well-known Eckmann-Hilton argument [5]. Below, we give an exact characterization of theories \( L \) for which tensoring with nondeterminism \( L_P \) is conservative.

For the sake of readability, we restrict the further development to **pointed** large Lawvere theories: since \( L_P \) has a constant (\( \emptyset \)), an evident necessary condition for \( L \to L \otimes L_P \) to be injective is that \( L \) can conservatively be made pointed, i.e. \( L \to L \otimes L_\bot \) (Definition 4.2) must
be injective — this is equivalent to \( L \) being already pointed if \( L \) has a constant, and a complex issue not in scope of the current investigation otherwise (one sufficient condition is that \( L \) is generated by equations having the same free variables on both sides). The main point is now that pointed large Lawvere theories carry a natural preorder:

**Definition 5.1** (Approximation). Let \( L \) be a pointed Lawvere theory. We compare elements of hom-sets \( L(n, m) \) under the **approximation preorder** \( \sqsubseteq \), which is the smallest preorder (strictly speaking: family of preorders on hom-sets) with \( \bot \) as a bottom element and closed under the rule

\[
(\pi_{\sqsubseteq}) \quad \forall i. \ [\sigma_i] \circ f \sqsubseteq [\sigma_i] \circ g \\
\overset{h \circ f \subseteq h \circ g}{\text{if } h \subseteq \text{domain of } f}
\]

(equivalently, tupling and composition are monotone).

Roughly, \( f \sqsubseteq g \) if \( f \) is obtained from \( g \) by repeatedly deleting subterms and applying the given equations.

**Example 5.2.** 1) The approximation ordering on \( L_P \) and its variants is the subset relation. More generally, the approximation ordering coincides with the induced ordering in any additive theory, see Lemma 5.3 below.

2) In the list monad, \( l \sqsubseteq k \) for lists \( l, k \) iff \( l \) can be obtained from \( k \) by deleting some of its entries.

3) The approximation ordering on the theory \( L_{\text{mult}} \) of weighted nondeterminism is multiset containment.

4) The approximation ordering on the partial state monad \( S \to (S \times \bot) \) is the extension ordering.

**Lemma 5.3.** Let \( L \) be an additive (hence pointed) large Lawvere theory. Then the approximation preorder \( \sqsubseteq \) on \( L \) coincides with the order \( \leq \) induced by the additive structure.

**Lemma 5.4.** Let \( L \) be a pointed Lawvere theory. Then the tensor map \( L \to L \otimes L_P \) preserves the approximation preorder.

All this indicates that the notion of conservativity should take into account the approximation preorder.

**Definition 5.5.** Let \( L \) be a pointed large Lawvere theory. We say that \( L \) **admits unbounded nondeterminism** if the tensor injection \( \sigma_1 : L \to L \otimes L_P \) is an order embedding, i.e. \( \sigma_1 \) is monic and **reflects the approximation ordering** in the sense that \( f \sqsubseteq g \) whenever \( \text{cl}(f) \subseteq \text{cl}(g) \).

That is, \( L \) admits unbounded nondeterminism if tensoring \( L \) with \( L_P \) is order-theoretically conservative.

For the remainder of this section, let \( L \) be a pointed large Lawvere theory. In a first step, we apply two key simplifications to the description of \( L \otimes L_P \) given by the representation according to (1) (p. 6): (1) implies that we can represent a morphism in \( L \otimes L_P \) as a tuple of sets of \( L \)-morphisms with some variables substituted by \( \varnothing \). We can, however

i) represent tuples of sets by sets of tuples using Cartesian products of sets, and

ii) get rid of occurrences of \( \varnothing \) in the bottom layer by dint of the fact that we already have \( \bot \).

Based on these observations, we arrive at construction of the tensor \( L \otimes L_P \) that can be proved correct independently of Theorem 3.7. To begin, we define a precursor of \( L \otimes L_P \), a category \( T_0 \) whose objects are sets and whose morphisms \( n \to m \) are subsets of \( L(n, m) \), with composition defined pointwise and identities \( \{\text{id}\} \). We have identity-on-objects functors \( \sigma^0_1 : L \to T_0 \) and \( \sigma^0_2 : L_P \to T_0 \) defined by \( \sigma^0_1(f) = \{ f \} \), and by \( \sigma^0_2(A_i) = \{ e \mid e : m \to n, e(i) \leq \} \).
$A_i$ for all $i$} for a morphism $(A_i) : n \to m$ in $L_P$, i.e. a family of $m$ subsets $A_i \subseteq n$. The category $T_0$ inherits a functor $I^0 : \text{Set}^{op} \to T_0$ from $L$ via $\sigma^0_1$; it is easy to see that under the axiom of choice (!), $I^0$ maps products to weak products.

We then define a relation $\approx$ on the sets $T_0(n, m)$ (strictly speaking: a family of relations on hom-sets) inductively as the smallest equivalence that contains all instances of the axiom scheme

\[
(\bot) \quad \{\bot_n, m\} \approx \emptyset \subseteq L(n, m)
\]

and is closed under the infinitary rule

\[
(\pi) \quad \forall i. [\alpha_i]A \approx_1 [\alpha_i]B \quad \therefore CA \approx_1 CB
\]

where $L$-morphisms (such as $[\alpha_i]$) are meant to convert to singleton sets when appropriate. We refer to $\approx$ as rectangular equivalence. We put $T = T_0/ \approx$, and obtain functors $I : \text{Set}^{op} \to T$, $\sigma_1 : L \to T$, $\sigma_2 : L_P \to T$ by prolongation along $T_0 \to T$.

**Theorem 5.6.** The category $T$ of sets of $L$-morphisms modulo rectangular equivalence as constructed above is the tensor product $L \otimes L_P$ of the pointed theory $L$ with unbounded nondeterminism $L_P$.

Similar results hold for tensoring with $L_{P^*}$ and for tensoring finitary theories with $L_{P_{<\omega}}$ or $L_{P_{\leq\omega}}$. Salient points in the proof are that while the tensor law does not hold in $T_0$, it does hold up to rectangular equivalence, and moreover that the tensor law justifies pointwise composition.

In $L$, we have morphisms $\Delta_i = \prod_{j \neq i} \delta_{ij} : n \to n$, where for $i, j \in n$, $\delta_{ij} : 1 \to 1$ equals $[1_d]$ if $i = j$ and $\bot$ otherwise.

**Lemma 5.7.** For $f : n \to m$, $g : m \to k$ in $L$, $fg \approx \{f \Delta_i g \mid i \in m\}$. \[\therefore\]

Since the right hand side of the above equivalence is a join in the tensor $L \otimes L_P$, order theoretic conservativity will imply that it is a join already in $L$. We proceed to develop a characterization of the tensor in terms of order-theoretic closures from this observation.

**Definition 5.8.** We say that $A \subseteq L(n, m)$ is closed if $P$ is downclosed and closed under the rule

\[
(\Delta) \quad \forall i. \ g \Delta_i h \in \text{cl}(A). \quad \therefore h \in \text{cl}(A).
\]

We denote the closure of $A \subseteq L(n, m)$, i.e. the smallest closed superset of $A$, by $\text{cl}(A)$. We write $\text{cl}(f)$ for $\text{cl}(\{f\})$.

**Lemma 5.9.** For $A, B \subseteq L(n, m)$, $A \approx B$ iff $\text{cl}(A) = \text{cl}(B)$.

Consequently, the tensor $L \otimes L_P$ can be regarded as having closed subsets of $L(n, m)$ as morphisms $n \to m$. The following is, then, more or less immediate.

**Theorem 5.10** (Order-theoretic conservativity). Let $L$ be a pointed large Lawvere theory with approximation preorder $\sqsubseteq$ as defined above, and let $\sigma_1 : L \to L \otimes L_P$ be the injection into the powermonad.

a) The following are equivalent:

(i) $\sigma_1$ reflects the approximation ordering (Definition 5.5)

(ii) For all $f : n \to m$, $g : m \to k$ in $L$, $fg$ is a least upper bound of $\{f \Delta_i g \mid i \in m\}$.

(iii) For all $f : n \to m$ in $L$, $\text{cl}(f)$ is the downset $f_\downarrow = \{g \mid g \sqsubseteq f\}$.

Theorem 5.10 immediately implies that all absolutely free theories, including input and output, admit unbounded nondeterminism. Moreover, the partial state monad $S \to (S \times \bot)_+$ admits unbounded nondeterminism, and indeed Lemma 5.9 allows a quick identification of the tensor as the nondeterministic state monad $S \to \mathcal{P}(S \times \bot)$ (this can also be obtained from the characterization of the state.
monad transformer as a tensor [11]). Every finitely additive finitary Lawvere theory admits unbounded nondeterminism. As a consequence, adding finite nondeterminism to a finitary theory is conservative if adding unbounded nondeterminism is conservative. Multisets do not admit nondeterminism, as the upper bound of \{a, \bot\} and \{\bot, a\} is not \{a, a\} but \{a\}. Similarly, lists do not admit unbounded nondeterminism, as \{a, b\} is not a supremum of \{a\} and \{b\}. In both cases, already injectivity of the tensor map fails by Remark 5.12.

6. Conclusion

We have proved the existence of tensors of large Lawvere theories for the case that one of the components is uniform. This implies in particular that one can always tensor with unbounded nondeterminism and with continuations, in the latter case improving a previous existence result [9]. We have then given a characterization of pointed theories that can be conservatively tensored with nondeterminism, which means precisely that one can assume such theories to be completely additive. Completely additive theories support a calculus for Kleene iteration, in generalization of classical Kleene algebra, and, e.g., admit a generalized form of the classical Fischer-Ladner encoding.

Although our results already have a quite order-theoretic flavour, an important issue for future research is whether similar results can be obtained in a domain-theoretic setting, using enriched Lawvere theories over a suitable category of cpos. Another direction for extending our results is to generalize them to enrichment over toposes, with a view to covering presheaf-based effects such as local state [22] or name creation [24].

References

Appendix

1. Proof of Theorem 3.7.

The proof is by explicit syntactic construction of the tensor product $L = L_1 \varotimes L_2$. To begin, we define a (not necessarily locally small) category $C$ on top of $L_1, L_2$ as follows. For $f \in \text{Hom}_{L_1}(k, m)$ and $g \in \text{Hom}_{L_2}(n, k)$ let $f \ast g$ be a synonym for the pair $(f, g)$. We agree to omit the subscript at $\ast$ if it is clear from the context. We also agree that $\ast$ binds weaker than composition. Let us define objects of $C$ to be sets and morphisms from $\text{Hom}(n, m)$ to be finite paths

$$\{ f_1 \ast g_1 \mid \ldots \mid f_k \ast g_k \}$$

adhering to the typing constraints: $n$ is the source of $g_k$, $m$ is the target of $f_1$, and for $i = 1, \ldots, k - 1$, the source of $g_{i+1}$ is the target of $f_i$. We often omit brackets for one-element paths.

The identity morphisms of $C$ are the empty paths, and composition is concatenation of paths. Clearly, $C$ is a category. On every hom-set of $C$ we define an equivalence relation $\sim$ as the equivalence generated by the clauses

$$\{ \ldots \mid [\text{id}] \ast [\text{id}] \mid \ldots \} \sim \{ \ldots \mid \ldots \},$$

$$\{ \ldots \mid f[\varepsilon] \ast g \mid \ldots \} \sim \{ \ldots \mid f \ast [\varepsilon]g \mid \ldots \}$$

and

$$\{ \ldots \mid f \ast (n' \otimes g) \mid (f' \otimes m) \ast g' \mid \ldots \}$$

$$\sim \{ \ldots \mid f(f' \otimes m') \ast (n \otimes g)g' \mid \ldots \}$$

where $f' : n \to n'$ and $g : m \to m'$. By construction, $\sim$ is a congruence on $C$, so that we have a quotient category $C_/\sim$. Using the fact that $L_2$ is uniform, we show that every morphism of $C_/\sim$ has a representative of the form

$$\{ f \ast g \mid c_{L_2}^n \ast [\text{id}] \}.$$  \hfill (2)

To that end, let us take any morphism $f$ of $C$. By attaching sufficiently many elements $[\text{id}] \ast [\text{id}]$ in the end of $f$ we ensure that the length of it is at least 2 and the last element of it is $[\text{id}] \ast [\text{id}]$. Then we successively apply the following reduction sequence, whose net effect is length-decreasing, as long as possible:

$$\{ f_1 \ast g_1 \mid f_2 \ast g_2 \mid \ldots \} \sim \{ f_1 \ast g_1 \mid (f_2 \otimes m_2)[u_{f_2}]c_{L_2}^{k_2} \ast g_2 \mid \ldots \}$$

$$= \{ f_1 \ast g_1 \mid (f_2 \otimes m_2)[(u_{f_2})c_{L_2}^{k_1} \otimes 1) \ast (k_1 \otimes [\text{id}])g_2 \mid \ldots \}$$

$$\sim \{ f_1 \ast g_1 \mid (f_2 \otimes m_2) \ast ((s \otimes m_2) \otimes [\text{id}]) \mid \ldots \}$$

$$= \{ f_1 \ast g_1 \mid (\hat{f}_2 \otimes m_2) \ast [\text{id}] \mid [u_{f_2}]c_{L_2}^{k_1} \ast g_2 \mid \ldots \}$$

$$\sim \{ f_1 \ast g_1 \mid (\hat{f}_2 \otimes m_2) \ast [\text{id}] \mid [u_{f_2}]c_{L_2}^{k_1} \ast g_2 \mid \ldots \}$$

$$= \{ f_1 \ast g_1 \mid (\hat{f}_2 \otimes m_2) \ast [\text{id}] \mid [u_{f_2}]c_{L_2}^{k_1} \ast g_2 \mid \ldots \}$$

Here, $f_i : k_i \to m_i, g_i : n_i \to k_i$ and $\hat{f}_2 : s \to 1$. At the last step of the reduction we obtain a pair of the form (2).

Let us define $I : \text{Set}^{op} \to C_/\sim$ by putting $I(n) = n$ for every set $n$ and $I(e) = [\varepsilon] \ast [\text{id}]$ for every set-function $e : n \to m$. We would like to prove that $I$ weakly preserves small products, i.e. the families $I(x_i) : \sum_i n_i \to n_i$ define weak small products in $C_/\sim$. Let $f_i : m \to n_i$ be a family of morphisms in $C_/\sim$. First we consider a special case when every $f_i$ is presentable by a one-element path, e.g. $f_i \sim g_i \ast h_i$. Since $L_2$ has all small products, there exists a morphism $h : m \to \sum_i k_i$ of $L_2$ such that for every $i$, $h_i = [\varepsilon_i]h$. Analogously, since $L_1$ has all small products, there exists a morphism $g : \sum_i k_i \to \sum_i n_i$ of $L_1$ such that for every $i$, $g_i[\varepsilon_i] = [\varepsilon_i]g$. The equality $f_i = I(x_i)[g \ast h]$, characterizing weak products, now follows from the diagram

```
```

whose two cells are commuting in $L_1$ and in $L_2$ respectively. In general, the $f_i$ might not be presentable by one-element paths, but as we have argued above, they must be presentable by paths of length 2. In particular, for every $i$, $f_i \sim g_i h_i$ where both $g_i : k_i \to n_i$ and $h_i : m \to k_i$ are one-element. As we have proved, there exists $h$ such that for every $i$, $h_i = I(x_i)h$. On the other hand, every $g_i I(x_i)$ is easily seen to be equivalent to a one-element path and therefore, there exists $g$ such that for every $i$, $g_i I(x_i)g = I(x_i)gh$ and we are done.

Let us prove that $C_/\sim$ is locally small. Since every morphism of $C$ has the form (2), every hom-set of $C_/\sim$ has at most as many equivalence classes as
are non-equivalent morphisms (2) in the corresponding hom-set of \( C \). Let us fix a pair \( \langle f \circ k \mid \gamma \rangle \in \text{Hom}_C(n, m) \). Let \( c = \text{Hom}_{L_2}(0, 1) \). By local smallness of \( L_2 \), \( s = \text{Hom}_{L_2}(n + c + 1) \) is a set, and thus \( s \in \text{Ob}(L_1) \). For every \( i \in k \), \( g_i = [\gamma_i] \) belongs to \( \text{Hom}_{L_2}(n + c + 1) \) and we denote by \( u : k \to s \) the induced index transformation. Let \( \beta : n + c \to s \) be the tupling morphism for the whole family \( \text{Hom}_{L_2}(n + c, 1) \). Then

\[
\langle f \circ k \mid \gamma \rangle \sim \langle f \circ [u] \beta \mid \gamma \beta \rangle \sim \langle f [u] \beta \beta \mid \gamma \beta \rangle.
\]

We have thus shown that every morphism of \( \text{Hom}_{C/\sim}(n, m) \) has a representative in the set \( \text{Hom}_{L_2}(n, s) \times \text{Hom}_{L_2}(n, s) \) and hence \( \text{Hom}_{C/\sim}(n, m) \) is also a set.

Let \( \approx \) be the smallest congruence on \( C \), containing \( \sim \) and closed under the rule:

\[
\forall i, [\gamma_i]f \approx [\gamma_i]g \implies f \approx g.
\]

We then have a canonical functor \( C/\sim \to C/\approx \), which equips \( C/\approx \) with all small weak products, which due to (3) are in fact products. By postcomposing \( I : \text{Set}^{op} \to C/\sim \) with the canonical projection, we obtain a product preserving functor \( I : \text{Set}^{op} \to C/\approx \), so that \( L = C/\approx \) is a Lawvere theory. We will be done once we show that \( L = L_1 \otimes L_2 \). We define functors \( \sigma_i : L_i \to L \) by

\[
\sigma_1(f) = f \circ [\text{id}], \quad \sigma_2(g) = [\text{id}] \circ g
\]

(omitting equivalence class formation from the notation). The following calculation ensures commutativity of \( \sigma_1 \) and \( \sigma_2 \):

\[
(\sigma_2(g) \otimes \eta')(m \circ \sigma_1(f)) = \langle ([\text{id}] \circ g) \otimes \eta'(m \circ \gamma) \rangle = \langle ([\text{id}] \circ g \otimes \eta') \otimes (m \circ \gamma \circ \text{id}) \rangle = \langle [\text{id}] \circ (m' \circ \gamma) \circ \text{id} \rangle = m' \circ \gamma & \text{id} = \langle \gamma \circ \text{id} \rangle \Rightarrow \langle \gamma \circ \text{id} \rangle
\]

Finally, let \( L' \) be another Lawvere theory equipped with a pair of commuting morphisms \( \alpha_i : L_i \to L' \). We define a morphism of categories \( \alpha : C \to L' \) to be identity on objects and by the equations \( \alpha\text{Id} = \text{id} \) and

\[
\alpha\langle f_1 \circ g_1 \mid \ldots \mid f_k \circ g_k \rangle = \alpha_1(f_1) \circ \alpha_2(g_1) \ldots \alpha_1(f_k) \circ \alpha_2(g_k)
\]
on morphisms. It is straightforward to verify by definition that \( f \approx g \) implies \( \alpha(f) = \alpha(g) \). Therefore, by the characteristic property of the quotient category, \( \alpha \) lifts to a morphism of Lawvere theories \( \alpha : L \to L' \). It is again easy to verify that for \( i = 1, 2, \alpha_i = \alpha \sigma_i \).

Uniqueness of \( \alpha \) is clear. Therefore \( L \) is indeed a tensor product of \( L_1 \) and \( L_2 \) and we are done.

\[ \square \]

## 2. Proof of Proposition 4.9

1) Let for every \( \Gamma \vdash t : TA \times TA, h(t) = \text{fst}(t) + \text{snd}(t) \). Then

if \( b \) then \( p \) else \( q \)

\[
= \text{case} \ b \text{ of } \text{inl} \Rightarrow p; \text{inr} \Rightarrow q
= \text{case} \ b \text{ of } \text{inl} \Rightarrow h(p, \bot); \text{inr} \Rightarrow h(\bot, q)
= h(\text{case} \ b \text{ of } \text{inl} \Rightarrow \langle p, \bot \rangle; \text{inr} \Rightarrow \langle \bot, q \rangle)
= h(\text{case} \ b \text{ of } \text{inl} \Rightarrow p; \text{inr} \Rightarrow \bot,
= \text{case} \ b \text{ of } \text{inl} \Rightarrow \bot; \text{inr} \Rightarrow q)
= \text{do} (\text{case} \ b \text{ of } \text{inl} \Rightarrow \text{ret} ; \text{inr} \Rightarrow \bot); p + \text{do} (\text{case} \ b \text{ of } \text{inl} \Rightarrow \text{ret} ; \text{inr} \Rightarrow \bot); q
= \text{do} (\text{case} \ b \text{ of } \text{inl} \Rightarrow \text{ret} ; \text{inr} \Rightarrow \bot); p + \text{do} (\text{case} \ b \text{ of } \text{inl} \Rightarrow \text{ret} ; \text{inr} \Rightarrow \bot); q
= \text{do} b?; p + \text{do} (\text{not} b)?; q
\]

and we are done.

2) First, note that by Lemma 5.3, \( \sqsubseteq \) coincides with \( \leq \). By part (i), we need to show that

\[
\text{do } x \leftarrow (\text{init } x \leftarrow \text{ret } x \text{ in } (\text{do } b?; p)^*); (\text{not } b)?; \text{ret } x
\]

is the least fixed point of

\[
q \mapsto \text{do } b?; x \mapsto p; q + \text{do } (\text{not } b)?; \text{ret } x
\]

(4) First observe that (4) is a fixed point of (5):

\[
\text{do } b?; x \mapsto p; (\text{do } x \leftarrow (\text{init } x \leftarrow \text{ret } x \text{ in } (\text{do } b?; p)^*) ; (\text{not } b)?; \text{ret } x)
= \text{do } x \leftarrow (\text{do } b?; x \mapsto p; \text{init } x \leftarrow \text{ret } x \text{ in } (\text{do } b?; p)^* + \text{ret } x); (\text{not } b)?; \text{ret } x
= \text{do } x \leftarrow (\text{do } x \leftarrow (\text{do } b?; p) \text{ in } (\text{do } b?; p)^* + \text{ret } x); (\text{not } b)?; \text{ret } x
= \text{do } x \leftarrow (\text{do } b?; p) \text{ in } (\text{do } b?; p)^* + \text{ret } x); (\text{not } b)?; \text{ret } x
\]

In order to show that (4) is the least fixed point, suppose \( q \) is some other fixed point of (5). Then

\[
\text{do } b?; x \mapsto p; q \leq q
\]

\[
\text{do } (\text{not } b)?; \text{ret } x \leq q
\]

From the former inequality, by (ind 2):

\[
\text{do } x \leftarrow (\text{init } x \leftarrow \text{ret } x \text{ in } (\text{do } b?; p)^*); q \leq q
\]
from which we conclude by the latter inequality,

$$\text{do } x \leftarrow \text{init } x \leftarrow \text{ret } x \in (\text{do } b?; p^*)^\ast; (-b)?; \text{ret } x \leq q.$$  

Therefore (4) is indeed the least fixed point of (5) and the proof is thus completed.  

3. Proof of Lemma 5.3

To prove that $\subseteq$ is contained in $\leq$, it suffices to show that $\leq$ has the closure properties defining $\subseteq$. By definition, $\subseteq$ has $\perp$ as bottom. To see that $\leq$ is closed under $(\pi_{\subseteq})$, let $f, g : n \to m$ in $L$ such that $[x_i]f \leq [x_j]g$ for all $i \in m$. By definition, this means that $[x_i]f + [x_j]g[x_i](f + g) = [x_j]g$ for all $i$, so that $f + g = g$, i.e. $f \leq g$; since by definition, composition is monotone w.r.t. $\leq$, it follows that $hf \leq hg$ for $h : m \to k$.

To show that, conversely, $\leq$ is contained in $\subseteq$, let $f \leq g$. Then $f = f + \perp \subseteq f + g = g$.  

4. A Direct Construction of the Nonempty Powermonad

To pave the ground for the direct construction of the power monad, i.e. the proof of Theorem 5.6, we describe the direct construction of the nonempty powermonad, i.e. tensoring with non-blocking unbounded nondeterminism.

We need a preliminary lemma to ease the proof of the tensor equation.

**Lemma A.1.** In the notation of Definition 3.1, the tensor equation reduces to the case $m_1 = m_2 = 1$.

**Proof:** We prove the general case as follows: To check commutation of the requisite diagram for arbitrary $m_1, m_2$, it suffices to check commutation for all postcompositions with the product projections $\pi_{ij} = \pi_i(m_1 \otimes m_2)$ where $\pi_i : m_1 \to 1$ and $\pi_j : m_2 \to 1$ are product projections. Therefore $\pi_{ij} : (f_1 \otimes m_2) = \pi_i(m_1 \otimes \pi_j)(f_1 \otimes m_2) = \pi_i(f_1 \otimes \pi_j)$ and $\pi_{ij} : (m_1 \otimes f_2) = \pi_j(\pi_i \otimes m_2)(m_1 \otimes f_2) = \pi_j(\pi_i \otimes f_2) = \pi_i \otimes \pi_j(f_2)$. (Note here that for $f : m \to n$ and a map $e : k \to l$, $f \otimes e : n \times l \to m \times k$ is definable as the morphism into the $k$-fold product $m \times k$ whose postcomposition with the $j$-th projection $m \times k \to m$ is $f|_{m_1 \otimes f_2}$ where $e(j)$ is the $e(j)$-th product projection $n \times l \to n$.) Next note that $(\pi_i \otimes \pi_j)(n_1 \otimes f_2) = \pi_i f_1(n_1 \otimes \pi_j f_2)$ and $(\pi_i \otimes \pi_j f_2)(f_1 \otimes n_2) = \pi_j f_2(\pi_i \otimes n_2)$, so that we are done by commutation of

$$n_1 \times n_2 \xrightarrow{n_1 \otimes \pi_j f_2} \xrightarrow{\pi_i f_1} n_1 \xrightarrow{\pi_j f_2} n_2 \xrightarrow{\pi_i f_1} n_1.$$  

Let $L$ be a Lawvere theory. We give a construction of the tensor $T = L \otimes L_{\pi}$. We begin by constructing a category $T_0$ with an identity-on-objects functor $I_0 : \text{Set} \to T_0$, with the same notation as for Lawvere theories, with the following properties:

- $I_0$ maps products to weak products;
- $T_0$ has functors $F^0 : L \to T_0$, $F_{\pi}^0 : L_{\pi} \to T_0$ that commute with the respective functors from $\text{Set}$

That is, $T_0$ will fail to be the tensor $L \otimes L_{\pi}$ on two counts: tupling morphisms need not be unique in $T_0$, and the two functors from the component theories into $T_0$ need not commute. One of the surprises in the construction is that repairing the first defect will remedy also the second one.

Morphisms $n \to m$ in $T_0$ are just nonempty sets of $L$-morphisms. Composition is defined by $AB = \{ab \mid a \in A, b \in B\}$; identities are singleton sets $\{\text{id}_n\}$. The functor $F^0 : L \to T_0$ maps a morphism $f$ to the singleton $\{f\}$. We then define $I_0$ as the composite $\text{Set} \to L \to T_0$. The tupling $\langle A_i \rangle : n \to \sum m_i$ of $k$ $T_0$-morphisms $A_i : n \to m_i$ is defined as

$$\langle A_i \rangle = \{\langle f_i \rangle \mid f_i \in A_i \text{ for all } i\},$$  

where $\langle f_i \rangle$ denotes tupling in $L$. We regard morphisms $n \to m$ in $L_{\pi}$ as $m$-tuples $(A_i)$ of nonempty subsets of $n$. Then the functor $F_{\pi}^0 : L_{\pi} \to T_0$ maps $(A_i)$ to

$$F_{\pi}^0((A_i)) = \{\langle e \rangle \mid e : n \to m, e(i) \in A_i \text{ for all } i\}.$$  

In the special case $m = 1$, in which case a morphism $n \to m$ is just a single subset $A \subseteq n$, note that $F_{\pi}^0(A) = \{[x_i] \mid i \in A\}$.

This completes the definition of $T_0$. We need to check a few properties:

- The composite $\text{Set} \to L_{\pi} \to T_0$ coincides with $I_0$. To see this, let $e : m \to n$ be a map. In $L_{\pi}$, $[e] : n \to m$ is the $m$-tuple $\{\langle e(i) \rangle \}_{i \in m}$. Under $F_{\pi}^0$, this becomes the set

$$\{\langle e \rangle \mid e : m \to n, e(i) \in \{e(i)\} \text{ for all } i\} = \{[e]\}.$$  

- The tupling morphisms project back to their components: For $A_i : n \to m_i$ in $T_0$, we have
\[ \pi_j(A_i) = \{ \pi_j(f_i) \mid (f_i) \in \prod A_i \} = \{ f_j \mid (f_i) \in \prod A_i \} = A_j, \]

as the \( A_i \) are nonempty (note that this requires the axiom of choice).

We now proceed to repair the mentioned defects by quotienting \( T_0 \) by an appropriate equivalence relation. We define the relation \( \approx \) as the smallest reflexive transitive relation closed under the infinitary rule

\[ (\pi) \forall i, [\kappa_i]A \cong [\kappa_i]B \Rightarrow CA \cong CB. \]

**Lemma A.2.** The relation \( \approx \) satisfies the following properties.

1) \( \approx \) is symmetric.
2) \( \approx \) is a congruence w.r.t. composition.
3) \( \approx \) is a congruence w.r.t. tupling.

**Proof:**

1) Put \( \approx^* = \approx \cap \approx^{-1} \) where \( \approx^{-1} \) denotes the inverse relation. Clearly, \( \approx^* \) is reflexive and transitive. Moreover, \( \approx^* \) is easily seen to be closed under \((\pi)\). Consequently, \( \approx \subseteq \approx^* \), so that \( \approx \) is symmetric.

2) Let

\[
A \approx^* B \iff \forall L, R. LAR \approx LBR.
\]

Then \( \approx^* \) is clearly reflexive and transitive. Moreover, \( \approx^* \) is closed under \((\pi)\): if \( [\kappa_i]A \approx [\kappa_i]B \) for all \( i \), then in particular \( [\kappa_i]AR \approx [\kappa_i]BR \) for all \( i, R \) and hence \( LAR \approx LBR \) for all \( L, R \). Therefore, \( \approx^* \) contains \( \approx \), so that \( \approx \) is a congruence.

3) Let \( A_i, B_i : n \to m_i \) and \( A_i \cong B_i \) for \( i \in k \). To prove \( \langle A_i \rangle \cong \langle B_i \rangle \), we have to show \( \langle \kappa_{(i,j)} \rangle \langle A_i \rangle \cong \langle \kappa_{(i,j)} \rangle \langle B_i \rangle \) for \( i, j \in k \). Now \( \kappa_{(i,j)} = (\kappa_i \times \kappa_j)_{i,j} \), where \( \kappa_i : m_i \to \sum m_i \) is the coproduct embedding (and hence \( \kappa_i \) is a weak product projection). Thus, we have \( \langle \kappa_{(i,j)} \rangle \langle A_i \rangle = \langle \kappa_i \rangle A_i \cong \langle \kappa_j \rangle B_i = \langle \kappa_{(i,j)} \rangle \langle B_i \rangle \), using the assumption \( A_i \cong B_i \) and the fact that \( \approx \) is congruent w.r.t. composition.

A last observation that needs to be made is that the tensor law holds in \( T_0 \) modulo \( \approx \): Let \( A \subseteq m \), corresponding to the morphism \( \bar{A} = \{ [\kappa_i] \mid i \in A \} : m \to 1 \) in \( T_0 \), and let \( f : n \to 1 \) in \( L \) (identified with a singleton in \( T_0 \)). We have to show that the diagram

\[
\begin{array}{ccc}
\pi_j(A_i) & \to & \langle f_i \rangle \\
\downarrow & & \downarrow \\
\prod A_i & \to & \prod A_i \\
\end{array}
\]

commutes. Now we have

\[
\begin{align*}
\bar{A} \otimes \bar{A} &= \langle [\kappa_i \times \kappa_m] \rangle_{i \in m} \\
&= \{ ([\kappa_i \times \kappa_m]_{j \in A}) \mid j \in A \}_{i \in m} \\
&= \{ [\kappa_{(i,j)}] \}_{i \in m} \mid (j) \in A^n \}
\end{align*}
\]

and hence

\[
f(\bar{A} \otimes \bar{A}) = \{ f([\kappa_{(i,j)}]) \mid (j) \in A^n \}.
\]

On the other hand, we have

\[
\bar{A}(f \otimes m) = \bar{A}(\bar{f} \otimes m) \\
= \{ [\kappa_{(i,j)}] \}_{i \in m} \mid j \in A \}
\]

so that equivalence of the two sides follows from

\[
\{ [\kappa_{(i,j)}] \}_{i \in m} \mid j \in A \} = \{ [\kappa_{(i,j)}] \}_{i \in m} \mid (j) \in A^n \}.
\]

To prove (7), we compare the projections along \( \kappa_{(i,j)} \), \( i \in n \), on both sides, and thus reduce the goal to the evident equality

\[
\{ [\kappa_{i}] \}_{i \in n} \mid j \in A \} = \{ [\kappa_{(i,j)}] \}_{i \in m} \mid (j) \in A^n \}.
\]

We have thus shown that \( T = T_0 / \approx \) is a morphism \( n \to 1 \) in \( L \), whose image under \( G \), we denote by \( \hat{A} : L \to S \), \( G \) satisfies the tensor law for \( G \) and \( G \). We define a functor \( \hat{G} : T \to S \) as follows. In preparation, we note that every subset \( A \subseteq n \) is a morphism \( n \to 1 \) in \( L \), whose image under \( G \), we denote by \( \bar{A} \subseteq L \). In particular, for a morphism \( A : n \to m \) in \( T_0 \) we have \( \bar{A} : (A \subseteq L) : (m, n) \to 1 \) in \( S \). Moreover, we have for each set \( A \) a morphism

\[
s_n = \hat{G}((\langle f \rangle)_{f \in L(n)}) : n \to L(n)
\]

in \( S \), where we denote by \( L(n) \) the action of the monad induced by \( L \), i.e. simply \( L(n) = L(n, 1) \). We then define a functor \( \hat{G}_0 : T_0 \to S \) by putting, for \( A : n \to m \) in \( T_0 \),

\[
\hat{G}_0(A) = \hat{A} s_n.
\]

In general, we then put

\[
\hat{G}_0(A) = \langle G([\kappa_i]A) \rangle_{i \in m}
\]

for \( A : n \to m \) in \( T_0 \) (noting that this agrees with the previous definition in case \( m = 1 \)).

To establish the requisite properties of \( \hat{G}_0 \), we need the following lemma.
Lemma A.3. Let \( A \subseteq n, \ B \subseteq m \), and let \( e : n \to m \) such that \( e[A] = B \). Then
\[
(A \subseteq n)[e] = (B \subseteq m)
\]
in \( S \).

Proof: Immediate from the corresponding equality in \( L_{P^*} \).

To begin, we now show that \( \bar{G}_0 \) preserves \( \llbracket \cdot \rrbracket \), which will then also imply that \( G_0 \) preserves identities. Thus, let \( e : n \to m \) be a map; we have to show that \( \bar{G}_0[e] = [e] \), which by applying product projections on both sides and by definition of \( \bar{G}_0 \) immediately reduces to the case \( n = 1 \), i.e. \( e = \varpi_j \) for some \( j \in m \). Now we have
\[
\bar{G}_0[\varpi_j] = \{ [\varpi_j] \}_{m = 1} = (1 \subseteq 1)[\varpi[\varpi_j]]_{m = m} = \varpi_j,
\]
where the second step is by Lemma A.3 (applied to \( \varpi[\varpi_j] \)) \( [\llbracket \cdot \rrbracket] \).

The crucial point in the proof is now to establish that \( \bar{G}_0 \) preserves composition. Again, this reduces immediately to the case where the codomain of the composite is 1. Thus, let \( A : n \to 1 \), and let \( B : k \to n \) in \( T_0 \), put \( B_j = [\varpi_j]B \) and \( \nu_j = \lambda_j . [\varpi_j]g : L(k, n) \to L(k) \) for \( j \in n \). By Lemma A.3, we then have \( B_j = B[\varpi_j] \). We start to transform \( \bar{G}_0AG_0B \):
\[
\bar{G}_0AG_0B = \hat{A}s_n(B_j s_k)_{\nu_j} = \hat{A}s_n(B[\varpi_j] s_k)_{\nu_j} = \hat{A}s_n(s \otimes B)([\varpi_j] s_k)_{\nu_j} = \hat{A}(L(n) \otimes \hat{B})(s \otimes L(k, n))([\varpi_j] s_k)_{\nu_j},
\]
using the tensor law in the last step. We proceed to analyse the right-hand subterm of the last term separately: we claim that
\[
s_n \otimes L(k, n)\langle [\varpi_j] s_k \rangle_{\nu_j} = G\langle f g \rangle_{f \in L(n), g \in L(k, n)}.
\]
We then note moreover that the right-hand side of (8) equals \( [c] s_k \), where \( c : L(n) \times L(k, n) \to L(k) \) is composition (this is proved by precomposing both sides with the projections \( [\varpi_{(f,g)}] \): we have \( [\varpi_{(f,g)}][c] s_k = [c, \varpi_{(f,g)}] s_k = \varpi_{(f,g)} s_k = G(f g) \)). We then conclude the argument by
\[
\hat{A}(L(n) \otimes \hat{B})(s_n \otimes L(k, n))([\varpi_j] s_k)_{\nu_j} = \hat{A} \times B[c] s_k = ABs_k,
\]
again using Lemma A.3 in the last step.

It remains to prove our claim (8). We note that \( n \times L(k, n) \) is the \( n \)-fold product of \( L(k, n) \) in \( S \), with projections \( p_j = [\lambda g . (j, g)] \) for all \( j \in n \), and at the same time the \( L(k, n) \)-fold product of \( n \), with projections \( q_g = [\lambda j . (j, g)] \) for all \( g \in L(k, n) \), and similarly \( L(n) \times L(k, n) \) is the \( L(k, n) \)-fold product of \( n \), with projections \( q_g = [\lambda f . (f, g)] \) for all \( g \in L(n) \). We then prove (8) by precomposing both sides with \( [\varpi_{(f,g)}] \). We have
\[
\begin{align*}
[\varpi_{(f,g)}] & (s_n \otimes L(k, n))([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j][\lambda f . (f, g)](s_n \otimes L(k, n))([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j]q_g(s_n \otimes L(k, n))([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j]s_n q_g([\varpi_j] s_k)_{\nu_j} \\
& = G(f)q_g([\varpi_j] s_k)_{\nu_j}.
\end{align*}
\]
Thus we are done once we show that \( q_g([\varpi_j] s_k)_{\nu_j} = G(g) \). To this end, we precompose both sides with \( [\varpi_j] \) and calculate
\[
\begin{align*}
[\varpi_j] & q_g([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j][\lambda j . (j, g)]([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j][\lambda g . (j, g)]([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j]q_j([\varpi_j] s_k)_{\nu_j} \\
& = [\varpi_j]([\varpi_j] s_k) \\
& = [\varpi_{\varpi_{(j,g)}}] s_k = G([\varpi_j] g) = [\varpi_j]Gg.
\end{align*}
\]
This concludes the proof that \( \bar{G}_0 \) preserves composition. It is then clear that \( \bar{G}_0 \) factors through \( T \), as its kernel satisfies all properties featuring in the inductive definition of \( \approx \) (the kernel is, of course, reflexive and transitive, and it is closed under \( \pi \) because tupling is unique in \( S \)). Uniqueness of the arising factorizing morphism \( \bar{G} : T \to S \) is clear, because every morphism \( n \to m \) in \( T \) has the form \( F_{P^*}(A)F(B) \), where \( B : n \to k \) in \( L \) and \( A : k \to n \) in \( L_{P^*} \).

Summarizing the above, we have shown that
\[
T_0/\approx \text{ is the tensor product of } L \text{ and } P^*.
\]
One consequence of this is the following property:

Lemma A.4. \( \approx \) is congruent w.r.t. union.

5. A Direct Construction of the Power Tensor

We proceed to give details for Theorem 5.6. The construction of tensor products with the full powerset theory \( L_P \) is similar to the one for the nonempty powerset, but more involved due to the fact that the full powerset theory has a constant, \( \emptyset \). The general construction of tensoring a theory \( L \) with uniform theories tells us that for such a case, we have to expect
a three-layered normal form that has operations of $L_P$ on top, under this a layer of operations of $L$, and at the bottom a layer consisting not only of variables but possibly also of occurrences of $\emptyset$.

To simplify matters, we have assumed that the given theory $L$ is pointed, with the unique constant denoted $\bot$. This will in particular allow us to replace occurrences of $\emptyset$ in the bottom layer with $\bot$, thus effectively reverting to a two-layered structure.

Under the assumption that $L$ is pointed, the tensor $L \otimes L_P$ is constructed as follows. As in the case of the nonempty power tensor, we begin by constructing a preliminary category $T_0$. Morphisms $n \to m$ in $T_0$ are (possibly empty) subsets of $L(n, m)$. Composition is pointwise, as previously; also, the definition of the embedding functors $L \to T_0$, $L_P \to T_0$ from the component theories remains unchanged, similarly for the indexing functor $\text{Set} \to T_0$. The crucial difference with the nonempty powerset theory is that we have $\mathcal{P}(\emptyset)$.

Next, we quotient $T_0$ by rectangular equivalence, i.e. the relation $\approx$ defined inductively as the smallest reflexive and transitive relation closed under rule $(\pi)$ and additionally satisfying the axiom

$$(\bot) \cup \{\bot_{n,m}\} \approx \emptyset$$

as well as the symmetric $\emptyset \approx \{\bot_{n,m}\}$ for all $n,m$. We have

**Lemma A.5.** The relation $\approx$ satisfies the following properties.

1) $\approx$ is symmetric.
2) $A \approx \tilde{A}$ for all $A$.
3) $\approx$ is a congruence w.r.t. composition.
4) $\approx$ is a congruence w.r.t. tupling.

**Proof:**

1) As in the nonempty case.

2) Trivial.

3) Let

$$A \approx^c B \iff \forall L, R. LAR \approx LBR.$$  

As in the nonempty case, $\approx^c$ is easily seen to be reflexive, transitive, and closed under $(\pi)$. It remains to see that $\approx^c$ contains all instances of $(\bot)$, i.e. (considering only one of the two symmetric cases of $(\bot)$) that for all $L, R$, $L\{\bot\}R \approx L \otimes R = \emptyset$. But this follows from pointness of $L$: for $l \in L$, $r \in R$, we have $llr = \bot$, and therefore $L\{\bot\}R = \{\bot\} \approx \emptyset$. Therefore, $\approx^c$ contains $\approx$, so that $\approx$ is a congruence.

4) Let $A_i, B_i : n \to m_i$ and $A_i \approx B_i$ for $i \in k$. To prove $\langle A_i \rangle \approx \langle B_i \rangle$, we have to show $[\kappa_{i,j}]\langle A_i \rangle \approx [\kappa_{i,j}]\langle B_i \rangle$ for $i \in k$, $j \in m_i$. Now $\kappa_{i,j} = (\kappa_j \times \text{id}_k)i_i$, where $i_i : m_i \to m = \sum m_i$ is the coproduct embedding. Thus, we have $[\kappa_{i,j}]\langle A_i \rangle = [\kappa_j]A_i \approx [\kappa_j]B_i \approx [\kappa_{i,j}]\langle B_i \rangle$, using the assumption $A_i \approx B_i$, congruence w.r.t. composition, and claim 2 of this lemma.

So far, we have established that $T := T_0/\approx$ is a large Lawvere theory that has theory morphisms $L \to T$ and $L_P \to T$. To prove that $T$ is a candidate for the tensor $L \otimes L_P$, we need to show that the tensor law holds. The argument is mostly as in the nonempty case: we need only check those cases where the empty set can occur within a tupling operation; the only case in point is where $\tilde{A} = \emptyset$, in the notation of (6). This case, however, is taken care of by the fact that $\emptyset \approx \{\bot\}$ and by pointness of $L$, which ensures that for $\{\bot\}$ in place of $\tilde{A}$, both paths in (6) equal $\{\bot\}$.

It remains to prove the universal property. Given a further candidate $S$, i.e. a large Lawvere theory with morphisms $G : L \to S$ and $G_P : L_P \to S$ satisfying the tensor law, we define $G_0 : T_0 \to S$ as before; the proof that $G_0$ respects composition and $[\Box]$ is unchanged from the nonempty case. Again, it is clear that $G_0$ factors through $T$ because its kernel satisfies the inductive definition of $\approx$, including all instances of $(\bot)$ as these are implied by validity of the tensor law in $S$. Uniqueness of the factorization is, again, clear.

**6. Proof of Lemma 5.9**

We need a preliminary lemma:
Lemma A.6. Let $A : n \to m$ in $T_0$. Then for all $a : n \to m$ and all $b : m \to k$,

$$a \in \cl(A) \implies ba \in \cl((bA)).$$

Proof: It suffices to show that the set $\bar{A} = \{a \in \cl(A) \mid ba \in \cl((bA))\}$ contains $A$, is downward closed, and is closed under $\Delta$. The first and second properties are clear; we check the third property. Thus, let $h : n \to k$ and $g : k \to m$ such that $g\Delta_i h \in \bar{A}$ for all $j \in \bar{m}$. Then $bh\Delta_i h \in \cl((bA))$ for all $j$, and hence $bgh \in \cl((bA))$, so that $gh \in \bar{A}$ as required. 

Proof of Lemma 5.9: In preparation, note that

$$\Delta_i = \langle \delta_{ij} \rangle_{j \in \bar{n}} [\varepsilon_i].$$

Only if: It suffices to show that the equivalence $\simeq$ defined by $A \simeq B$ iff $\cl(A) = \cl(B)$ is closed under $\Delta$ and contains all instances of $\bot$. The latter holds by the definition of $\cl(\bot)$. To check the former, we first show that $\bot$ is left congruent w.r.t. composition. Thus, let $A, B : n \to m$ and let $C : m \to k$ such that $\cl(A) = \cl((CB))$. Then $\cl((CB))$ is downward closed and closed under $\Delta$, it suffices to prove $\cl((CB)) \subseteq \cl((CB))$. Thus let $c \in C, a \in A$. Then $a \in \cl((CB))$ by assumption, and therefore $[\varepsilon_i] \subseteq \cl((CB))$ by Lemma A.6. It follows that $\cl((CA)) \subseteq \cl((CB))$. The converse implication is shown symmetrically.

It remains to show that $\bot$ is closed under $\Delta$. Thus, let $A, B : n \to m$ such that $\cl([\varepsilon_i]A) = \cl([\varepsilon_i]B)$ for all $i \in \bar{m}$. We have to show $\cl(A) = \cl(B)$. By (10) and Lemma A.6, we have $\cl_{\Delta_i a} \subseteq \cl((\Delta_i B))$ for all $i \in \bar{m}, a \in A$. By downward closedness, $\cl((\Delta_i B)) \subseteq \cl(B)$, so that we obtain $a \in \cl(B)$ by rule (A). It follows that $\cl(A) \subseteq \cl(B)$; the reverse inclusion is shown symmetrically.

If: It suffices to show that for $A : n \to m$ in $T_0$, $A \simeq \cl(A)$. Since $\simeq$ is congruent w.r.t. set union, it suffices to show that $A \approx A \cup \{f\}$ for all $f \in \cl(A)$, which will follow if we show that the set

$$\bar{A} := \{f : n \to m \mid A \approx A \cup \{f\}\}$$

(which clearly contains $A$) is downward closed and closed under $\Delta$.

- $\bar{A}$ is downward closed: define a preorder $\preceq$ by $f \preceq g : \iff \{f, g\} \approx g$; then $\preceq$ is easily seen to be closed under $\simeq$, and hence contains $\subseteq$. Now let $g \in \bar{A}, f \sqsubseteq g$. Then $f \preceq g$ and therefore $A \cup \{f\} \approx A \cup \{g\} \approx A$ using congruence w.r.t. union.

- $\bar{A}$ is closed under $(\Delta)$: Let $f : n \to m, g : m \to k$, and let $f\Delta_i g \in \bar{A}$ for all $i \in \bar{m}$. Then $A \cup \{fg\} \approx A \cup \{f\Delta_i g \mid i \in \bar{m}\} \approx A$, using congruence w.r.t. composition and Lemma 5.7.

7. Proof of Lemma 5.7

By rule $(\pi)$, it suffices to prove $[\varepsilon_i]g \approx [\varepsilon_i] \{\Delta_i g \mid i \in \bar{m}\}$ for all $j \in \bar{m}$. But the right hand side of this equivalence equals $[\varepsilon_i]g \cup \{\bot\}$, which is equivalent to $[\varepsilon_i]g$ by $(\bot)$ and congruence w.r.t. union.

8. Proof of the Order-Theoretic Conservativity
Theorem 5.10

1) $\Rightarrow b):$ Immediate from Lemma 5.7.

$b) \Rightarrow c)$: The inclusion $\cl(f) \subseteq f_\bot$ holds because under $\bot$, $f_\bot$ is closed under $\Delta$. The reverse inclusion holds because $\cl(f)$ is downclosed.

$c) \Rightarrow a):$ Immediate by Lemma 5.9.

2) Immediate from property $c)$.

9. Details for Remark 5.12

Let $L$ be simply ordered. We prove that if $\sigma_1 : L \to L \otimes L$ is injective, then it reflects the ordering. According to Theorem 5.10, we have to show that for $f : n \to m, g : m \to k$ in $L$, $fg = \bigsqcup_{i \in \bar{m}} f\Delta_i g$. Thus $h$ be an upper bound of $\{f\Delta_i g \mid i \in \bar{m}\}$. Since $L$ is simply ordered, there exist $f', g'$ such that $f'g' \subseteq h$, $f'g'$ is a minimal upper bound of $\{f\Delta_i g \mid i \in \bar{m}\}$, and for every $j$ there is $i$ such that $f'\Delta_i g \subseteq f\Delta_i g$. Hence $\cl(fg) = \cl(f'g')$. Injectivity of $\sigma_1$ then implies that $fg = f'g' \subseteq h$.

10. Details for Example 5.13

Let $L$ be a finitely additive finitary Lawvere theory; we prove that $L$ admits unbounded nondeterminism. Recall that by Lemma 5.3, the approximation ordering on $L$ coincides with the ordering induced by the additive structure. By Theorem 5.10, all that remains to be shown is that for all $f : n \to m, g : m \to k$ in $L$, $fg = \sum_{i \in \bar{m}} f\Delta_i g$, where we can restrict to finite $n, m, k$ because $L$ is finitary. Since finite sums commute with composition, this reduces to showing that $\sum_{i \in \bar{m}} \Delta_i = \id_m$, which is straightforward by comparing projections.