A Divertimento on MonadPlus and Nondeterminism

Tarmo Uustalu

Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia

Abstract

In the Haskell community, there is a controversy about what the laws of the MonadPlus type constructor class ought to be. We suggest that there is no single universal correct answer, however there is a universal method. Important classes of notions of finitary nondeterminism are captured by what we call monads of semigroups and monads of monoids, but also by monads of different specializations of semigroups and monoids. Some of these specializations of monads are exotic and amusing too.

Keywords: monads, algebraic operations, semigroups, nondeterminism

1. Introduction

Monads are a widely used in functional programming, especially Haskell, as an abstraction of computational effects. The mathematics of monads is very well understood, as they are a central concept of category theory. But Haskell’s standard library features also a more specific type constructor class MonadPlus to cover a particular variety of effects, namely different notions of finitary nondeterminism. Oddly, the mathematics of MonadPlus has been unclear. The question of what the “right” laws of MonadPlus ought to be has been open from the beginning and there is no commonly agreed answer. Moreover, until the recent work by Rivas et al. [14], there have been no attempts to justify any proposal by mathematical considerations.

In this note, we pursue one principled way of addressing this question. We take inspiration from the algebraic operations approach of Plotkin and Power [13]. We suggest that there is no single universal correct answer, rather there is a universal method. Important classes of notions of finitary nondeterministic choice are captured by what we call monads of semigroups and monads of monoids, but also by monads of different specializations (varieties) of semigroups and monoids. We argue that such special types of monads, uniformly defined following a certain pattern, are well-motivated, since they enjoy neat mathematical properties. The appropriate special type of monads can be chosen following a clear method, once it is fixed what idea of finitary nondeterminism is had in mind, i.e., what exactly is meant to be observable about nondeterministic
computations. Some of these specializations of monads are quite exotic, but all of them are instructive and fun to study.

I have titled this piece a divertimento, since I consider it lighthearted; certainly it was intended so. I have tried to make it entertaining, yet serious, and thus fit for a festive social function like a birthday celebration. And above all I hope I found a beautiful theme that is easy to memorize. I have learned a lot from you, José Nuno, and the thing I value most about the interactions we have had is your overflowing enthusiasm about beautiful mathematics and music.

The paper is organized as follows. First, in Section 2, we review the MonadPlus class of Haskell. In Section 3, we introduce monads of semigroups and show that this is a well-motivated concept with neat mathematical properties that also corresponds to a natural class of notions of finitary nondeterminism. In Section 4, we introduce monads of monoids as a similar concept with analogous properties. In Section 5, we discuss monads of some specializations of semigroups to capture narrower classes of notions of nondeterminism (where less and less operational detail is observable). In Section 6, we point to the closest related work, to conclude in Section 7.

We assume that the reader is knowledgeable about monads and their application to programming with effects.

To keep the presentation simple, especially concerning examples, we use Set as the base category throughout the presentation. But any category with finite products would work for most of our purposes.

2. The MonadPlus class of Haskell

The Control.Monad module of Haskell’s standard library provides a type class called MonadPlus [3]. Its definition says that a MonadPlus instance is a Monad instance equipped with two polymorphic functions mzero and mplus.

class Monad m => MonadPlus m where
  mzero :: m a
  mplus :: m a -> m a -> m a

  MonadPlus is thus a subclass of Monad. It is intended as an abstraction of notions of finitary nondeterminism rather than of notions of just any type of effectful computation. mzero is the operation of nullary choice (choice with no options) and mplus is binary choice. But MonadPlus has also been applied to organize search with backtracking, sometimes regarded as “don’t know” nondeterminism, in opposition to normal (demonic) “don’t care” nondeterminism.

The most important intended MonadPlus instance is the list monad, with the empty list and concatenation of lists as mzero and mplus. Other wanted instances include, e.g., the combination of the list monad with the state monad for some state set. The maybe monad is seen by some as an intended instance, as a monad to use for backtracking, while others want to exclude it, perhaps arguing that it should belong to a different type class with member functions of the same types but subject to a different set of laws.

A remarkable controversy about MonadPlus is that there is no agreement about the “right” laws (this question has been discussed in countless mailing list threads, wiki pages etc., see, e.g., [18, 5, 4]). It seems generally accepted that mzero and mplus should obey the laws of a monoid, i.e., the left and right unital laws and associativity:
mzero 'mplus' c == c
c 'mplus' mzero == c
(c 'mplus' c') 'mplus' c'' == c 'mplus' (c' 'mplus' c'')

Also, mzero should be a left zero of bind and bind should right distribute (i.e., distribute from the right) over mplus\(^3\), which is to require that \((>>= k)\) is a monoid homomorphism:

\[
mzero >>= k == mzero
(c 'mplus' c') >>= k == (c >>= k) 'mplus' (c' >>= k)
\]

(Right distributivity fails for the maybe monad.\(^4\))

Some require in addition that mzero be a right zero and bind left distribute over mplus:

\[
c >>= \_ -> mzero == mzero
c >>= \_ -> (k a 'mplus' k' a) == (c >>= k) 'mplus' (c >>= k')
\]

But left distributivity fails already for the list monad, since list concatenation is non-commutative.\(^5\)

Among further laws that one may want to impose for some purposes are commutativity and idempotence of mplus.

\[
c 'mplus' c' == c' 'mplus' c
c 'mplus' c == c
\]

Of course these fail for the list monad, as list concatenation is neither commutative nor idempotent. But lists may be used as representations for finite sets and under this view commutativity and idempotence hold on the level of denotations.

Yet another law sometimes suggested is the following so-called left catch law.

\[
\text{return } a 'mplus' c == \text{return } a
\]

This is satisfied by the maybe monad, but falsified by the list monad.\(^6\)

Together with right distributivity, left catch gives

\[
k a 'mplus' (c >>= k) == k a
\]

(just use that \(k a == \text{return } a >>= k\), which one may consider unreasonably strong.

In this paper, we aim to give a well-motivated method for answering the question of what the laws of the MonadPlus class should be. We concentrate first on the case where only mplus is present and justify a relatively weak axiomatization (only associativity together with as little more as one can feasibly get away with—it turns out that there are very good reasons to include right distributivity). Then we add mzero. And then we look at a number of laws that can be included optionally in order to capture narrower classes of notions of nondeterminism.

---

\(^3\)Confusingly, this law is called left distributivity in most documents about MonadPlus.

\(^4\)For \(k b = \text{if } b \text{ then Nothing else Just } ()\), one has \((\text{Just True} 'mplus' \text{Just False}) >>= k == \text{Nothing while (Just True} >>= k) 'mplus' (\text{Just False} >>= k) == \text{Just ()}.\)

\(^5\)For both \(k = k' = \text{return}\), one has \([x,y] >>= \_ -> (k a 'mplus' k' a) == [x,x,y,y] while ([x,y] >>= k) 'mplus' ([x,y] >>= k') == [x,y,x,y].\)

\(^6\)\text{return } x 'mplus' xs == x : xs, but \text{return } x == [ x ].
3. Monads of semigroups

We follow the algebraic operations approach of Plotkin and Power [13]. We take interest in monads that “support” some algebraic operations.

We say that a monad of semigroups is a monad \( T = (T, \eta, \mu) \) with a family of maps \( \oplus_X : TX \times TX \to TX \) natural in \( X \) such that \( \oplus \) is associative and \( \mu \) is right distributive over \( \oplus \):

\[
\begin{align*}
(TX \times TX) \times TX & \xrightarrow{\oplus_X \times TX} TX \times TX \\
TX \times (TX \times TX) & \xrightarrow{T \oplus X} TX \\
TX \times TX & \xrightarrow{\oplus_X} TX \\
T(TX) & \xrightarrow{\mu_X} TX \\
\end{align*}
\]

\( \alpha \) for \( X, Y, Z \in X \):

\[
\begin{align*}
\alpha_{TX,TX,TX} & : (TX \times TX) \times TX \to TX \\
\oplus_X & : TX \times TX \to TX \\
\end{align*}
\]

\( \mu(c \oplus c') = \mu c \oplus \mu c' \)

The right distributivity law can also be expressed in terms of \((-)^*\) (the Kleisli extension operation) rather than \( \mu \) and then ensures also naturality of \( \oplus \):

\[
k^* (c \oplus c') = k^* c \oplus k^* c'
\]

The paradigmatic example of a monad of semigroups is the nonempty list monad \( (TX = X^+, \oplus_X = +) \). We will shortly see that this example is special—the nonempty list monad is the initial monad of semigroups. Some other obvious instances are the nonempty finite powerset monad and the list monad.

We claim that monads of semigroups are canonical in several ways and make a good abstraction of those notions of finitary nondeterminism based on binary choice where the temporal order of binary choices in a computation (under the view of a computation as a binary leaf-labelled tree of choices with values at the leaves, this corresponds to the interior of the tree) is not observable.

It is immediate from the definition that a monad of semigroups \( (\mathbb{T}, \oplus) \) delivers semigroups in the sense that, for any \( X \), \( (TX, \oplus_X) \) is a semigroup—this is ensured by the associativity law. Moreover, right distributivity says that \( \mu_X \) is a semigroup homomorphism between \( (T(TX), \oplus_T X) \) and \( (TX, \oplus_X) \).

Abstractly, while a monad is nothing but a monoid in the monoidal category \( ([\text{Set}, \text{Set}], \times, \text{id}, \cdot) \), a monad of semigroups is the same as a right near-semiring without additive unit in the right near-semiring category without additive unit \( ([\text{Set}, \text{Set}], \times, \text{id}, \cdot) \).

---

7A right near-semiring without additive unit is a set \( R \) with an additive semigroup structure \( (\cdot) \) and a multiplicative monoid structure \( (\cdot, \cdot) \) such that multiplication right distributes over addition, i.e., \( (r + r') \cdot s = r \cdot s + r' \cdot s \). Note that, differently from a semiring without zero, commutativity of addition and left distributivity of multiplication over addition are not required. A right near-semiring also has an additive unit 0 that is a left zero for multiplication, i.e., \( 0 \cdot s = 0 \). A right near-semiring category (without or with additive unit) compares to a monoidal category like a right near-semiring (without or with additive unit) compares to a monoid.

8In the standard terminology, cf. Golan [7], a semiring is a ring without negatives but with zero, so we have chosen to emphasize ‘without additive unit’ here. But in an alternative terminology, a ring without negatives is a rig, and a semiring is a ring with neither negatives nor zero.
This becomes evident, when we write the laws of $\oplus$ on the level of functors:

$$
\begin{array}{c}
(T \times T) \times T \overset{\oplus \times T}{\longrightarrow} T \times T \\
\alpha_{T,T,T} \downarrow \\
T \times (T \times T) \overset{\oplus}{\longrightarrow} (T \times T) \cdot T \overset{\oplus \times \mu}{\longrightarrow} T \times T
\end{array}
$$

But the more interesting properties of a monad of semigroups pertain to its Eilenberg-Moore category and Kleisli category. Also, the initial monad of semigroups (i.e., the initial object in the category of monads of semigroups) stands out by additional special properties.

The Eilenberg-Moore category of a monad of semigroups

First, algebras of a monad of semigroups $(T, \oplus)$ are semigroups in the sense of there being a functor $F : \text{EM}(T) \to \text{Sgrp}$ preserving carriers:

$$
\begin{array}{c}
\text{EM}(T) \overset{F}{\longrightarrow} \text{Sgrp} \\
U \downarrow \\
C
\end{array}
$$

The semigroup structure $\boxplus : A \times A \to A$ that $F$ associates to a $T$-algebra $(A, \alpha)$ is defined by

$$
\boxplus = A \times A \overset{\eta_A \times \eta_A}{\longrightarrow} TA \times TA \overset{\oplus_A}{\longrightarrow} TA \overset{\alpha}{\longrightarrow} A
$$

On maps, $F$ is identity.

The converse is also true. Any monad $T$ whose algebras are semigroups, i.e., that comes with a functor $F$ as above, is a monad of semigroups. For any $X$, the map $\oplus_X$ is given by the semigroup structure corresponding to the $T$-algebra $(TX, \mu_X)$ (the free $T$-algebra on $X$). Right distributivity holds, i.e., $\mu_X$ is a semigroup homomorphism between $(TX \oplus TX, \oplus_{TX})$ and $(TX, \oplus_X)$, because $\mu_X$ is a $T$-algebra homomorphism between $(TX, \mu_X)$ and $(TX, \mu_X)$.

The correspondence between semigroup structures on a monad $T$ and carrier-preserving functors from $\text{EM}(T)$ to $\text{Sgrp}$ is a bijection. (And further, this bijection extends to an isomorphism of categories.)

The Kleisli category of a monad of semigroups

The Kleisli category of a monad of semigroups $(\mathbb{T}, \oplus)$ has a semigroup structure $\oplus_{X,Y}$ on every homset $\text{Kl}(\mathbb{T})(X,Y)$ given by

$$
(X \overset{k}{\longrightarrow} T \ Y) \oplus_{X,Y} (X \overset{k'}{\longrightarrow} T \ Y) = X \overset{(k,k')}{\longrightarrow} TY \times TY \overset{\oplus_Y}{\longrightarrow} TY
$$

This does not however mean enrichment of $\text{Kl}(\mathbb{T})$ over $\text{Sgrp}$. It is the case that

$$
\ell \bullet (k \oplus k') = (\ell \bullet k) \oplus (\ell \bullet k')
$$
(it is a consequence of right distributivity). But
\[(k \oplus k') \cdot \ell = (k \cdot \ell) \oplus (k' \cdot \ell)\]
fails in general (for that, one would also need left distributivity of \(\mu\) over \(\oplus\)). Only a special case holds:
\[(k \oplus k') \cdot Jf = (k \cdot Jf) \oplus (k' \cdot Jf)\]
(Here \(\circ\) is Kleisli composition and \(J : \mathsf{Set} \to \mathsf{Kl}(T)\) is the left adjoint of the Kleisli adjunction.)

It is possible, of course, to insist on enrichment, but then the properties of the Eilenberg-Moore category change. We have chosen to not pursue this line here.

**The initial monad of semigroups**

The initial monad of semigroups \((T_0, \oplus_0)\) has special properties.

First, it is the free semigroup delivering monad, i.e., for any \(X\), \((T_0 X, \oplus_{0X})\) is the free semigroup on \(X\). The free semigroup on a set \(X\) is given by the set of nonempty lists over \(X\) with concatenation (as nonempty lists serve as unique normal forms of leaf-labelled trees under associativity). So \(T_0 X = X^+\).

But also, for this monad of semigroups, the carrier-preserving functor \(F_0\) from the Eilenberg-Moore category to the category of semigroups becomes an isomorphism. The \(T_0\)-algebra structure \(\alpha : T_0 A \to A\) for a semigroup \((A, \boxplus)\) is given by the unique semigroup morphism from the free semigroup \((T_0 A, \oplus_{0A})\) to \((A, \boxplus)\) sending \(\eta_{0A}\) to \(\text{id}_A\):

\[
\begin{array}{ccc}
A & \xrightarrow{\eta_{0A}} & T_0 A & \xleftarrow{\oplus_{0A}} & T_0 A \times T_0 A \\
& & \downarrow{\alpha} & & \downarrow{\alpha \times \alpha} \\
& & A & \xleftarrow{\oplus} & A \times A
\end{array}
\]

As monads with isomorphic Eilenberg-Moore categories are isomorphic, the initial monad of semigroups is the unique monad with this property.

Besides the nonempty list monad, monads of semigroups include, for instance, the nonempty multiset monad \((T X = M_T^* X)\), the nonempty finite powerset monad \((T X = P_T^0 X)\), the list monad \((T X = X^+)\), the combination of the nonempty list monad and the state monad for some state set \((T X = S \Rightarrow (S \times X)^+)\) etc.

We refrain from demonstrating this here, but there is also an obvious concept of strong monads of semigroups, exhibiting useful properties. All set functors are uniquely strong and all natural transformations between them are strong, hence this concept only becomes relevant in more general settings.

### 4. Monads of monoids

Similarly to monads of semigroups, one can define and study monads of monoids.

We say that a *monad of monoids* is a monad of semigroups \((T, \oplus)\) with a family of maps \(\alpha_X : 1 \to T X\) natural in \(X\) such that \(\alpha\) is both left and right unital wrt. \(\oplus\) and

\[
\begin{array}{ccc}
A & \xrightarrow{\eta_{0A}} & T_0 A & \xleftarrow{\oplus_{0A}} & T_0 A \times T_0 A \\
& & \downarrow{\alpha} & & \downarrow{\alpha \times \alpha} \\
& & A & \xleftarrow{\oplus} & A \times A
\end{array}
\]
also a left zero of $\mu$ (the latter equation is “right distributivity” of $\mu$ over $\circ$):

\[
\begin{array}{ccc}
1 \times TX & \xrightarrow{\sigma_X \times TX} & TX \times TX \\
\downarrow{\lambda_{TX}} & & \downarrow{\rho_{TX}} \\
TX & \xrightarrow{\oplus_X} & TX
\end{array}
\quad
\begin{array}{ccc}
TX \times 1 & \xrightarrow{TX \times \sigma_X} & TX \times TX \\
\downarrow{\oplus_X} & & \downarrow{\sigma_{TX}} \\
TX & \xrightarrow{\rho_{TX}} & TX
\end{array}
\quad
\begin{array}{ccc}
1 & \xrightarrow{1} & 1
\end{array}
\]

$\circ \oplus c = c$

$c = c \oplus \circ$

$\mu \circ = \circ$

Expressed in terms of $(-)^*$ rather $\mu$, the left zero law says

$k^* \circ = \circ$

and ensures also that $\circ$ is natural, so this does not need to be required separately.

Everything we stated about monads of semigroups in the previous section applies also to monads of monoids, suitably adjusted. In particular, a monad of monoids is the same thing as a right near-semiring (with additive unit) in the right near-semiring category (with additive unit) ([\textbf{Set}, \textbf{Set}], $1$, $\times$, Id, $\cdot$). And a monad is a monad of monoids if and only if its algebras are monoids (in the sense of there being a carrier-preserving functor from the Eilenberg-Moore category of the monad to the category of monoids).

The initial monad of monoids is the free monoid delivering monad. The free monoid on $X$ is given by the set of all lists over $X$, including the empty list. Thus the initial monad of monoids is defined by $T_0X = X^*$.

Monads of monoids account for those notions of nondeterminism where also nullary choice is allowed whereby the temporal order in which choices are made in a computation (the interior of the computation tree) is unobservable. In the case of the initial monad of monoids, all of the rest (the leaf labels with their order and multiplicity) is observable.

For the initial monad of monoids, $\circ$ is also a right zero of $\mu$:

\[
\begin{array}{ccc}
T1 & \xrightarrow{\tau_1} & 1 \\
\downarrow{T \circ X} & & \downarrow{\circ_X} \\
T(TX) & \xrightarrow{\rho_X} & TX
\end{array}
\]

$\mu(T \circ c) = \circ$

or, in terms of $(-)^*$:

$(\lambda_\circ)^* c = \circ$

But from our perspective this is little more than a coincidence of lucky circumstances. Not every monad of monoids has this property, initiality helps. For example, this equation fails for the monad sending any $X$ to the set of non-wellfounded nullary-binary leaf trees over $X$, quotiented by the least congruence containing associativity and the unital laws.

Indeed, take as $c$ any such tree with an infinite number of labelled leaves. Applying $(\lambda_\circ)^*$ to $c$ replaces all labelled leaves of $c$ with $\circ$. But it is not possible to rewrite this tree into $\circ$ using the right and left unital laws of $\oplus$ only a finite number of times.
5. Monads of special classes of semigroups

There is no reason why we should confine our interest to semigroups or monoids. Depending on what we take finitary nondeterminism to mean (i.e., what information we want to be observable about a computation in the sense of a binary or a nullary-binary leaf-labelled tree), we may want to replace semigroups with other types of algebras with one binary operation, typically semigroups of some variety (semigroups satisfying additional equations), or algebras with a nullary and a binary operation, typically monoids of some variety. The recipe remains exactly the same as for semigroups: the algebraic operations and equations have to be stated on the level of functors and in addition one has to stipulate right distributivity of multiplication over the algebraic operations.

Monads of commutative semigroups

The equations of most interest are commutativity and idempotence. We begin with commutativity.

We say that a monad of commutative semigroups is a monad of semigroups 
\((T, \oplus)\) such that \(\oplus\) is commutative, i.e.,

\[
\begin{array}{c}
TX \times TX \\
\oplus_X
\end{array} 
\xrightarrow{\sigma_{TX,TX}} 
\begin{array}{c}
TX \times TX \\
\oplus_X
\end{array}
\]

\[c \oplus c' = c' \oplus c\]

All that we stated in Section 3 about monads of semigroups carries over to monads of commutative semigroups, mutatis mutandis.

The initial monad of commutative semigroups \(T_0\) delivers free commutative semigroups. The free commutative semigroup on \(X\) is the set of finite nonempty multisets over \(X\). So \(T_0 X = M^{>0}_f X\).

Other monads of commutative semigroups include, e.g., the finite nonempty powerset monad \((TX = P^{>0}_f X)\), the finite multiset monad \((TX = M_f X)\) and the combination of the finite nonempty multiset monad with the state monad \((TX = S \Rightarrow M^{>0}_f (S \times X))\).

Monads of commutative semigroups stand for notions of nondeterminism where, besides the temporal order of choices, also the order of outcomes is unobservable. In the case of the initial monad of commutative semigroups, everything else can be observed.

The initial monad of commutative semigroups satisfies left distributivity:

\[
\begin{array}{c}
 T (TX \times TX) \\
 T (TX)
\end{array} 
\xrightarrow{T \oplus X} 
\begin{array}{c}
 T (TX) \times T (TX) \\
 \mu_X \times \mu_X
\end{array}
\xrightarrow{\mu_X} 
\begin{array}{c}
 T X \times TX \\
 \oplus_X
\end{array} 
\xrightarrow{\mu_X} 
\begin{array}{c}
 T X \\
 \oplus_X
\end{array}
\]

\[\mu (T (\oplus) c) = \mu (T \text{fst} c) \oplus \mu (T \text{snd} c)\]

or, in terms of \((-)^*\):

\[(\lambda x. k x \oplus k' x)^* c = k^* c \oplus k'^* c'\]
But similarly to the case of monads of monoids and the right zero law, this is “incidental”: left distributivity does not hold for all monads of commutative semigroups. For example, it fails for the monad of commutative semigroups that sends any set $X$ to the set of all non-wellfounded binary leaf trees over $X$, quotiented by the least congruence containing associativity and commutativity. The intuitive reason is that the inductively defined congruence cannot deal with an infinite number of swaps.

**Monads of bands and semilattices**

We can similarly define that a *monad of bands* [resp. semilattices] is a monad $(T, \oplus)$ of semigroups [resp. commutative semigroups] such that $\oplus$ is idempotent, i.e.,

\[
\begin{array}{c}
TX \\
\oplus_X
\end{array} \xrightarrow{\Delta_T X} \begin{array}{c}TX \times TX \\
\oplus_X
\end{array}
\]

Again, everything we stated about monads of semigroups earlier carries over to monads of bands and monads of semilattices, suitably adjusted.

The initial monad of bands delivers free bands. The free band on a set $X$ is given by the set of square-free nonempty lists over $X$. So $T_0 X = \{xs : X^+ \mid xs \text{ is square-free}\}$. A nonempty list is said to be square-free, if no nonempty list occurs in it as a sublist twice in a row. The free band over a set $X$ arises as the quotient of the set of nonempty lists over $X$ by the least congruence containing idempotence, and square-free nonempty lists make canonical representatives for the equivalence classes by serving as unique normal forms.

The notion of nondeterminism corresponding to the initial monad of bands may feel artificial and is perhaps not very useful: the order of outcomes is observable, multiplicity is unobservable for adjacent blocks of outcomes, but otherwise observable. The reason is, of course, that, idempotence alone only allows one to collapse adjacent occurrences of a sublist. Still, in semigroup theory, free bands are a natural object of study.

The initial monad of semilattices delivers free semilattices. The free semilattice on a set $X$ is the nonempty finitary powerset of $X$, i.e., the set of all nonempty finite subsets of $X$. So this special monad is given by $T_0 X = \mathcal{P}^{\leq 0} X$. This corresponds to the probably most popular notion of nondeterminism: besides the temporal order of choices both the order and multiplicity of outcomes are unobservable.

**More exotic special types of monads**

There is nothing saying we should stop at commutativity and idempotence. There are many other interesting equations that lead to meaningful notions of nondeterminism.

We can define a *monad of right commutative semigroups* as a monad of semigroups
where $\oplus$ is right commutative:

$$TX \times (TX \times TX) \xrightarrow{T \times \sigma_{TX,TX}} TX \times (TX \times TX)$$

$$\oplus_X^{(3)} \quad \oplus_X^{(2)}$$

$$TX \quad TX \times \sigma_{TX,TX}$$

$$TX \times (TX \times TX) \xrightarrow{T \times \sigma_{TX,TX}} TX \times (TX \times TX)$$

$$c \oplus (c' \oplus c'') = c \oplus (c'' \oplus c')$$

where $\oplus^{(3)}$ is the ternary version of $\oplus$ (i.e., $\oplus^{(3)}_X = \oplus_X \circ (TX \times \oplus_X)$).

Right commutativity is weaker than commutativity: two summands can only be reordered in the presence of some left context.

The initial special monad of this type is given by $T_0 X = X \times M_T X$. This corresponds to a notion of nondeterminism where the leftmost outcome of a computation is singled out, but the order of the other outcomes is unobservable. This may perhaps be relevant, e.g., in some parsing context: one might want to be able to single out the first parse of a string against a grammar as primary, but consider all other parses to have the same secondary status.

(Notice that in the presence of a left unit $o$ for $\oplus$, right commutativity becomes equivalent to commutativity and is thus no longer interesting on its own.)

Likewise we can define a *monad of left regular bands* as a monad of bands satisfying left regularity:

$$TX \times TX \xrightarrow{\Delta_{TX,TX}} (TX \times TX) \times TX \xrightarrow{\sigma_{TX,TX}} TX \times (TX \times TX) \xrightarrow{T \times \sigma_{TX,TX}} TX \times (TX \times TX)$$

$$\oplus_X \quad \oplus_X^{(3)}$$

$$TX \quad TX \times \sigma_{TX,TX}$$

$$TX \times (TX \times TX) \xrightarrow{T \times \sigma_{TX,TX}} TX \times (TX \times TX)$$

$$c \oplus (c' \oplus c) = c \oplus c'$$

The free left regular band on a set $X$ is given by the set of duplicate-free nonempty lists over $X$, i.e., nonempty lists where no element of $X$ may occur twice (anywhere, not just adjacently). The reason is that idempotence and left regularity together allow one to remove any duplicate sublist occurrences, whether adjacent or not. But merely removing single duplicate elements achieves the same. The addition of the semigroup is concatenation; if a duplicate arises in the concatenation of two duplicate-free nonempty lists, the leftmost of the two occurrences is kept.

Accordingly, the initial monad of left regular bands is given by $T_0 X = \{xs : X^+ \mid xs$ is duplicate-free$\}$. The multiplication of the monad is concatenation of a duplicate-free nonempty list of duplicate-free nonempty lists; if duplicates arise (now they can be many), the leftmost occurrence of an element is kept. This monad corresponds to a notion of nondeterminism where only the leftmost occurrence of any outcome is observable.

Notice that this monad can also be called the monad of nonempty total orders: duplicate-free nonempty lists over $X$ are representations of nonempty total orders formed of elements of $X$. 

10
A monad of rectangular bands is definable as a monad of bands satisfying the equation

$$TX \times (TX \times TX) \xrightarrow{T \times \text{snd}} TX \times TX \xrightarrow{T \times \text{snd}} TX \times TX \xrightarrow{\oplus_X} TX$$

$$c \oplus (c' \oplus c'') = c \oplus c''$$

The free rectangular band on $X$ is $X \times X$, as doubleton lists can be used as unique normals forms of nonempty lists under idempotence and rectangularity. Idempotence allows one to expand a singleton list to a doubleton listing the same element twice. The rectangularity equation (together with associativity, of course) allows one to drop all but the first and last elements of a list of at least two elements.

The initial monad of rectangular bands is thus given by $T_0 X = X \times X$. It corresponds to the notion of nondeterminism where only the leftmost and rightmost outcomes are observable.

The underlying monad of this monad of rectangular bands is isomorphic to the reader monad for $B$ (the booleans), given by $T_0 X = B \Rightarrow X$. The addition $\oplus_0$ is defined by $(c \oplus_0 c') \mathbf{ff} = c \mathbf{ff}$, $(c \oplus_0 c') \mathbf{tt} = c' \mathbf{tt}$.

As a final nontrivial example, we can consider monads of semigroups satisfying the additional equation

$$TX \times (TX \times TX) \xrightarrow{T \times \text{fst}} TX \times TX \xrightarrow{T \times \text{fst}} TX \xrightarrow{\oplus_X} TX$$

$$c \oplus c' = c \oplus c''$$

The initial special monad of this type is given by $T_0 X = X \times B$. It is the writer monad for $(B, \mathbf{ff}, \lor)$ with $\oplus_0$ defined by $(c, b) \oplus_0 (c', b') = (c, \mathbf{tt})$. The notion of nondeterminism is this: of multiple outcomes of a nondeterministic computation, only the leftmost is observable, but in addition one can observe whether there were further outcomes.

We finish with a degenerate example. A monad of left zero semigroups are is a monad of semigroups satisfying the equation

$$TX \times TX \xrightarrow{\oplus_X} TX \xrightarrow{\text{fst}} TX$$

$$c \oplus c' = c$$

Note that this equation is a definite description (definition) of $\oplus$. A monad of left zero semigroups is therefore really the same as a monad, as any monad is uniquely equipped with a monad of left zero semigroups structure. The initial monad of left zero semigroups
is the initial monad, i.e., the identity monad \((T_0 X = X)\). The corresponding notion of nondeterminism is that only the leftmost outcome of a computation is observable and we do not even get to know whether there were any further outcomes. Having an opportunity to choose between the left and the right is essentially no choice; you could say that the left is deterministically chosen for you.

6. Related work

Hinze [8] and Martin and Gibbons [12] were probably the first authors to discuss the laws of MonadPlus in research articles.

Rivas et al. [14] have recently identified MonadPlus instances with what we have here called monads of monoids and observed that they are right near-semirings (with additive unit) in the right near-semiring category (with additive unit) of endofunctors wrt. the product additive structure and composition multiplicative structure. They also identified instances of Alternative, which is a subclass of Applicative, with right near-semirings in a different right near-semiring category of endofunctors (with the product additive structure and Day tensor multiplicative structure).

In a different context—of collection types (databases) rather than nondeterminism—monads of monoids were introduced under the name of ringads as early as in 1990 by Wadler [17]. Wadler’s note was not published; the earliest published article on ringads is by Trinder [16]. Gibbons [6] has recently revisited ringads in the same context. For his purposes, he considered dropping the associativity condition on addition.

The algebraic operations approach, that we have built upon in this paper, is from Plotkin and Power [13].

Goncharov et al. [2] and Coumans and Jacobs [1] have studied additive monads, defined as monads whose Kleisli category is enriched over commutative monoids.

Manes and Mulry [11] described the initial monad of rectangular bands as an example (under the name of the rectangular bands monad), noting that the Eilenberg-Moore category of this monad is isomorphic to the category of rectangular bands.

Semiring categories were introduced by Laplaza [10] and Kelly [9] (under the name of ring categories, sic!).

Varieties of semigroups and monoids have been studied thoroughly in algebra; for a survey, see [15].

7. Conclusion

There are several small lessons to learn from what we have shown in this paper.

We have shown that monads of semigroups and monads of monoids are well-motivated mathematizations of the idea behind Haskell’s MonadPlus constructor type class. But it is equally meaningful to work with monads of some specific variety of semigroups or monoids, if a more narrow class of notions of finitary nondeterminism is intended. There is more freedom for creativity here than meets the eye. The operations of binary (and optionally nullary) choice attached to a monad must be subjected to the equations of the chosen specialization of semigroups or monoids and also to right distributivity of multiplication.
Seeking a concrete description for the initial one among the monads of semigroups or monoids of some variety, it is a good idea to check whether the free such semigroups or monoids admit unique normal forms.

Acknowledgements. We are grateful to the anonymous reviewers for the friendly and useful feedback and to Jeremy Gibbons for information on the early 1990s Glasgow work on ringads.

The author was supported by the ERDF funded Estonian centre of excellence project EXCS and ICT R&D support project Coinduction, the Estonian Ministry of Education and Research institutional research grant IUT33-13 and the Estonian Science Foundation grant no. 9475.

References


