LFCS

Laboratory for Foundations of Computer Science
Department of Computer Science - The University of Edinburgh

Tail Recursion Through Universal Invariants

by

C. Barry Jay

LFCS Report Series **ECS-LFCS-92-205**

LFCS
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

April 1992

Tail Recursion Through Universal Invariants

C. Barry Jay *
LFCS, Department of Computer Science
University of Edinburgh
The King's Buildings, Mayfield Road
Edinburgh, UK, EH9 3JZ
e-mail: cbj@dcs.ed.ac.uk

April 6, 1992

Abstract

Tail recursive constructions suggest a new semantics for datatypes, which allows a direct match between specifications and tail recursive programs. The semantics focusses on loops, their fixpoints, invariants and convergence. Convergent models of the natural numbers and lists are examined in detail, and, under very mild conditions, are shown to be equivalent to the corresponding initial algebra models.

1 Introduction

Tail recursion is a central feature of program construction because of its efficiency, but is usually assigned a secondary place in semantics, which is dominated by primitive recursion as expressed through initial algebras. The success of this approach is testimony to the ease with which we can use initial algebras to specify functions, and their theoretical power. The difficulty is that whenever such a specification is to be translated into code there remains the need to optimise it, often by conversion into tail recursive form. Conversely, it is not at all easy to provide the semantics of an existing tail recursive program in the initial algebra style. The solution is to interpret tail recursion directly, by giving central importance to the program (or function) loop and its properties.

Consider the problem of coding up the function which maps a pair (x, y) of natural numbers to $x + y + 1$. In ML, the natural numbers can be defined by

$$\texttt{datatypenat} = \texttt{zero} \mid \texttt{succ of nat}$$

*Research supported by The Royal Society of Edinburgh/BP

Using a primitive recursive programming style we obtain

$$\begin{aligned} \texttt{fun} \quad & \texttt{addsp(x,zero)} && = \texttt{succ(x)} \\ \mid \quad & \texttt{addsp(x,succ(y))} && = \texttt{succ(addsp(x,y))} \end{aligned}$$

whereas a tail recursive style yields

$$\begin{aligned} \texttt{fun} \quad & \texttt{addst(x,zero)} && = \texttt{succ(x)} \\ \mid \quad & \texttt{addst(x,succ(y))} && = \texttt{addst(succ(x),y)} \end{aligned}$$

in which the call to `addst` only occurs at the end of the recursion loop.

The definition of `addsp` can be interpreted directly using the initial natural numbers object (see Section 4.2) in the usual way. That `addst` represents the same function must then be established by induction.

The alternative is to interpret `addst` directly. Define a program `once` which performs one step of the unfolding of `addst` if it occurs, and is fixed otherwise.

$$\begin{aligned} \texttt{fun} \quad & \texttt{once(x,zero)} && = \texttt{(x,zero)} \\ \mid \quad & \texttt{once(x,succ(y))} && = \texttt{(succ(x),y)} \end{aligned}$$

Thus `once` is an endomorphism or, more descriptively, a *loop* on `nat * nat`. Furthermore `addst(once(x,y)) = addst(x,y)` which is to say that `addst` is an *invariant* for this loop. This can be exhibited more clearly by transforming the code of `addst` into the following form.

$$\begin{aligned} \texttt{fun} \quad & \texttt{addst2(x,zero)} && = \texttt{succ(x)} \\ \mid \quad & \texttt{addst2(x,y)} && = \texttt{addst2(once(x,y))} \end{aligned}$$

The first line of code, however, still performs the two roles of exiting from the loop and then acting on the result, of which only the former involves tail recursion. The pure recursion is captured by

$$\begin{aligned} \texttt{fun} \quad \texttt{conv(x,y)} \; & = \texttt{if once(x,y) = (x,y) then (x,y)} \\ & \qquad \texttt{else conv(once(x,y))} \end{aligned}$$

with the final result given by

$$\begin{aligned} \texttt{fun} \quad \texttt{addsc(x,y)} \; & = \texttt{let val (z,-) = conv(x,y) in succ(z)} \\ & \quad \texttt{end} \end{aligned}$$

Abstracting away from the particularity of the loop `once` we can define the function `converge` which acts on loops on equality types as follows

$$\begin{aligned} & \texttt{fun} \quad \texttt{converge(f)(x:''a)} && = \texttt{if f(x) = x then x else converge(f)(f(x));} \\ & \texttt{val} \quad \texttt{converge} && = \texttt{fn : (''a -> ''a) -> ''a -> ''a} \end{aligned}$$

Informally, the interpretation of this program is to iterate `f` until it becomes fixed, and then return the fixpoint. This suggests that tail recursion on equality types can always be expressed as `converge(f)` for some loop `f`. Of course, `converge(f)` may not terminate, and one of the goals of semantics is to determine when it will. Let the semantics of `f` be a loop f on C. The unbounded iteration of f can be interpreted as constructing the quotient $\mathtt{inv}(f) : C \to C_0$ of C under the equivalence relation $x \equiv f(x)$, that is to say, the *universal invariant* of f or, more formally, the categorical colimit of the loop f. Then to extract from each equivalence class a fixpoint of f is to provide another function $m : C_0 \to C$ such that

$$\begin{aligned} \mathtt{inv}(f) \circ m &= \mathtt{id}_{C_0} \\ f \circ m &= m \end{aligned}$$

Thus m picks out representatives of the equivalence classes which are fixed by f. Such a loop f is called *convergent.*

For example, the obvious set-theoretic interpretation of `once` is convergent, with universal invariant given by addition $N \times N \to N$ and $m(x) = (x, 0)$.

Convergence of loops can play a central role in proving the termination of programs, as has been demonstrated for while-loops in c.p.o.'s [12]. The current goal, however, is to characterise the natural numbers and lists in terms of the convergence of certain classes of loops, of which `once` is typical for the natural numbers. The following example typifies the situation for lists.

Consider the reverse operation on lists. The primitive recursive style yields the ML program

```
fun   revp[ ]       = [ ]
    | revp(h :: t)  = revp(t)@[h]
```

which takes $O(n^2)$ steps. The tail recursive approach first introduces an auxillary function `aux : 'a list * 'a list -> 'a list` as follows

```
fun   aux(l,[ ])     = l
    | aux(l,h :: t)  = aux(h :: l,t);
fun   revt(l)        = aux([ ],l)
```

which takes $O(n)$ steps. From the viewpoint of initial algebra semantics this program appears rather ad hoc. It is, however, the exact analogue of the construction of the reverse operation in the tail recursive semantics (see Section 5.2) which is built using the loop `shunt` on `'a list * 'a list` defined by

```
fun   shunt(l,[ ])     = (l,[ ])
    | shunt(l,h :: t)  = (h :: l,t)
```

The structure of the paper is as follows. The convergent loops can be defined in any category, and so are presented first, in Section 2. The natural setting in which to discuss lists, however, is a distributive category $\mathcal{D}$, in which products distribute over coproducts. Examples include the bicartesian closed categories familiar from domain theory and, since the results do not use the projections or diagonal of the product, the symmetric monoidal closed categories with cartesian coproducts, as used in linear logic. Such categories are also closed under the construction of the Kleisli category $\mathcal{D}_{\mathbf{T}}$ for a commutative monad $\mathbf{T}$ and so typically include categories of partial maps and of relations. Note that the Kleisli category over a cartesian closed category need have neither cartesian products (in the usual sense) nor exponentials. Other fundamental categories, of topological spaces, metric spaces, abelian groups, etc. are also distributive.

Section 4 introduces the convergent natural numbers object, proves a few basic arithmetic results, and shows it equivalent to an initial natural numbers object. Section 5 introduces the convergent list objects. The main result is that a convergent list object is an initial list object: the converse holds if there is a natural numbers object. Of course, this implies the corresponding result for the natural numbers (which are lists on the unit object): the natural numbers have been treated separately in view of their independent interest, and the simpler proofs obtained by ignoring the reverse operation, which is central to the list result.

It follows that the results apply in the Kleisli category $\mathcal{D}_{\mathbf{T}}$ of a commutative monad $\mathbf{T}$ on $\mathcal{D}$. Further, the free functor into such a Kleisli category preserves list objects, so that if $\mathcal{D}$ has all list objects then so does $\mathcal{D}_{\mathbf{T}}$.

Section 6 shows that the list construction is functorial, and indeed forms a commutative monad. Finally, Section 7 looks at termination, a stronger notion of convergence which requires an explicit numerical bound on the number of iterations required to reach a fixpoint. This yields the terminating list objects, which, when definable, are shown equivalent to the previous list concepts.

Much remains to be done. Future papers will address: general recursion in an abstract setting; lazy datatypes, and; matrices. The explication of general datatypes in the distributive setting is also of ongoing interest; see, for example, the work of Cockett [7], Walters [37, 17] and Kelly [20].

2 Loops

2.1 Fixpoints and Invariants

Let $\mathcal{C}$ be any category. A *loop* f on an object C in $\mathcal{C}$ is an endomorphism $f : C \to C$. A *loop morphism* from f to $g : D \to D$ is a morphism $h : C \to D$ such that $h \circ f = g \circ h$

$$\begin{tikzcd} C \arrow[loop left, "f"] \arrow[r, "h"] & D \arrow[loop right, "g"] \end{tikzcd}$$

These form a category $\mathcal{C}^c$ of *loop diagrams* whose composition and identities are inherited from $\mathcal{C}$. Thus there is a forgetful functor $U : \mathcal{C}^c \to \mathcal{C}$ which maps the loop f to its underlying object C. Loop diagrams are sometimes known as *dynamical systems* [27]. Fix a loop f on C for the rest of this section.

Note that although $\mathcal{C}^c$ can be regarded as a sub-category of the arrow category $\mathcal{C}^{\rightarrow}$ of $\mathcal{C}$ it is not a *full* sub-category since a morphism from f to $g : D \to D$ in $\mathcal{C}^{\rightarrow}$ consists of a *pair* of morphisms, $h_1 : C \to D$ and $h_2 : C \to D$ such that $h_2 \circ f = g \circ h_1$.

A fixpoint for f is often thought of as an element $x \in C$ such that $f(x) = x$. Categorically this generalises to say that a morphism $x : X \to C$ is *fixed* by f if $f \circ x = x$. That is, x is a cone for the loop diagram f. Thus the universal cone or limit of the loop, if it exists, is its *fixed subobject* or *fixpoints*

$$\begin{tikzcd} \mathtt{Fix}(f) \arrow[r, "\mathtt{fix}(f)"] & C \arrow[loop right, "f"] \\ X \arrow[u, dashed, "y"] \arrow[ur, "x"'] & \end{tikzcd}$$

which in **Sets** is just the subset of C of all fixpoints.

Of course the fixpoints can also be described as a special form of equaliser, namely of the parallel pair $f, \mathtt{id}_C : C \to C$ but it may easily happen that every loop in a category has a fixed object without every parallel pair having an equaliser.

For example, let $\mathbf{Pos}(\omega)$ be the category of ω-complete partial orders (c.p.o.'s) and continuous functions. Although this category is complete, its full sub-category $\mathbf{Pos}_\perp(\omega)$ whose objects are c.p.o's with least element $\perp$ does not have all equalisers, since the subset of points where the two functions agree need not even be inhabited, much less have a least element. It does, however, have fixed objects for loops since if f is continuous then the least element of $\texttt{Fix}(f)$ is $\bigsqcup f^n \bot$.

Dualising leads us to consider a cocone for a loop diagram, that is, a morphism $g : C \rightarrow Q$ such that $g \circ f = g$. These are called the *invariants* for f since their value is unchanged by applications of f. The colimit map $\texttt{inv}(f) : C \rightarrow \texttt{Inv}(f)$ of the loop is thus its *universal invariant* and $\texttt{Inv}(f)$ is its *invariant object* or invariants.

$$\begin{tikzcd} C \arrow[loop left, "f"] \arrow[r, "\texttt{inv}(f)"] \arrow[dr, "g"'] & \texttt{Inv}(f) \arrow[d, dashed, "h"] \\ & Q \end{tikzcd}$$

That is, if $g : C \rightarrow Q$ is an invariant for f then there is a unique morphism $h : \texttt{Inv}(f) \rightarrow Q$ such that $h \circ \texttt{inv}(f) = g$.

In **Sets** the universal invariant for f can be constructed as the quotient of C by the equivalence relation generated by $x \equiv f(x)$, that is,

$$x \equiv y \text{ iff there are natural numbers } j \text{ and } k \text{ such that } f^j(x) = f^k(y) \qquad (1)$$

For example, let $s : N \rightarrow N$ be the successor function on the natural numbers and consider the loop $k : N \times N \rightarrow N \times N$ (corresponding to once(succ) above) defined by

$$\begin{aligned} k(x, 0) &= (x, 0) \\ k(x, s(y)) &= (s(x), y) \end{aligned}$$

Its fixpoints are the pairs $(x, 0)$ and we may choose $\texttt{fix}(k)$ to be $N \times 0 : N \rightarrow N \times N$. One invariant is the parity comparator $N \times N \rightarrow \{\mathbf{even}, \mathbf{odd}\}$ which maps (x, y) to $\texttt{true}$ if x and y have the same parity, and to $\texttt{false}$ otherwise. If we construct the universal invariant as a quotient $\texttt{Inv}(k) = N \times N / \equiv$ then the equivalence class of (x, y) contains a unique fixpoint $(x + y, 0)$ so that addition $N \times N \rightarrow N$ is another representation of the universal invariant. For example, the parity comparator factors through addition since x and y have the same parity iff $x + y$ is even.

The successor s itself has no fixpoints. On the other hand if $g : N \rightarrow X$ is an invariant for s then g must be a constant function, i.e. factor through the unique map $N \rightarrow 1$ which latter is thus its universal invariant.

The two examples above show a loop k whose invariants correspond to its fixpoints (and so converges) and another s whose sole equivalence class contains no fixpoint (and so enters an "infinite loop"). These phenomena appear together in the following example. Define $t : N \to N$ by

$$t(n) = \texttt{if } n < 3 \texttt{ then } 0 \texttt{ else } n + 2$$

Then $\texttt{inv}(t) : N \to \{0, \mathsf{odd}, \mathsf{even}\}$ can be defined by

$$\begin{aligned} \texttt{inv}(t)(n) = \texttt{if } n < 3 \quad & \texttt{then } 0 \texttt{ else} \\ & \texttt{if } n \text{ is odd } \texttt{then odd else even} \end{aligned}$$

where 0 is the fixpoint and odd and even represent the infinite loops.

Now consider some loops in $\mathbf{Pos}(\omega)$. Since it is complete and cocomplete [29] the universal invariants all exist, but the general construction is complicated.

Let ω be the free chain $0 < 1 < 2 < \ldots < n < \ldots < \infty$. The successor loop s on ω has a unique fixpoint ∞. Its invariant object also has a unique element since continuity implies $\texttt{inv}(\infty) = \bigsqcup_n \texttt{inv}(n) = \texttt{inv}(0)$.

Let $N_\perp$ be the flat natural numbers (obtained by adding a least element to the discretely ordered set of natural numbers). The loop h on $N_\perp \to N_\perp$ defined by

$$\begin{aligned} h(g)(\perp) &= \perp \\ h(g)(0) &= 1 \\ h(g)(s(n)) &= s(n) * g(n) \end{aligned}$$

has a unique fixpoint given by the factorial function $fact : 1 \to (N_\perp \to N_\perp)$ and has 1 as its invariant object too.

2.2 Convergent Loops

The universal invariant of f identifies all those values which will be mapped to the same fixpoint, if there is one, but provides no clue as to what it might be, as can be seen from the loops above whose invariant object is 1. The additional information is supplied by a morphism $m : \texttt{Inv}(f) \to C$ which picks out the desired fixpoint. Then `converge(f)` can be interpreted by $m \circ \texttt{inv}(f) : C \to C$.

Definition 2.1 *A loop f on C in $\mathcal{C}$* converges *to a morphism $m : C_0 \to C$ if f has a universal invariant $\texttt{inv}(f) : C \to C_0$*

$$\begin{tikzcd} C \arrow[loop left, "f"] \arrow[rr, shift left, "\texttt{inv}(f)"] && C_0 \arrow[ll, shift left, "m"] \end{tikzcd}$$

and the following equations hold

$$\begin{aligned} \texttt{inv}(f) \circ m &= \texttt{id}_{C_0} \\ f \circ m &= m \end{aligned}$$

The two equations assert that m is a monomorphism that picks out representatives of the equivalence classes generated by $\mathtt{inv}(f)$, and that m is fixed by f. In the examples of the previous sub-section k converges to $N\times 0 : N \to N\times N$ while the successor s in **Sets** and t are not convergent. In $\mathbf{Pos}(\omega)$ the successor on ω converges to ∞ and h converges to the factorial function.

Lemma 2.2 *If f converges to $m : C_0 \to C$ and has a fixed subobject $\mathtt{fix}(f) : \mathtt{Fix}(f) \to C$ then $m = \mathtt{fix}(f)$ as a subobject of C.*
Proof Let $m = \mathtt{fix}(f) \circ n : C_0 \to C$ be the factorisation induced by the fixedness of m. The inverse of n will be given by $\mathtt{inv}(f) \circ \mathtt{fix}(f)$. Observe that

$$\mathtt{fix}(f) \circ n \circ \mathtt{inv}(f) \circ \mathtt{fix}(f) = m \circ \mathtt{inv}(f) \circ \mathtt{fix}(f) = \mathtt{fix}(f)$$

and stripping off the monomorphism $\mathtt{fix}(f)$ shows that $n \circ \mathtt{inv}(f) \circ \mathtt{fix}(f) = \mathtt{id}$. The other equation is immediate. □

It is easy, however, to construct a category with a convergent loop which doesn't have a fixed subobject. Previously [13, 14] the definition of convergence also assumed that $m = \mathtt{fix}(f)$. While aesthetically pleasing, such a symmetry between limits and colimits is not justified since, unlike the universal invariant, m is a semantic device for proving properties of programs, whose universality as a fixed subobject is unnecessary, either conceptually or in the proofs below. Indeed it was the difficulty of proving universality in the monoidal setting that lead to these observations. Further evidence is supplied by the following proposition, which is surely false for the symmetric definition.

Proposition 2.3 *Left adjoints preserve convergent loops.*
Proof Left adjoints preserve all colimits, including universal invariants, and also preserve equations. □

The next two propositions provide plenty of examples of convergent loops.

Proposition 2.4 *In* **Sets** *a loop f on C converges iff there is a function $n : C \to N$ such that for all $x \in C$ the element $f^{n(x)}(x)$ is fixed by f.*
Proof If f converges then the equivalence class of x in C contains a fixpoint x_0. Thus by (1) there is a least number $n(x)$ such that $f^{n(x)}(x) = x_0$ which is fixed. Conversely, the mapping taking x to $f^{n(x)} \in \mathtt{Fix}(f)$ is the universal invariant. □

The existence of such an n in the above proposition asserts that the loop f *terminates*. That is, that there is an explicit bound on the number of iterations of f required to reach a fixpoint (see Section 7). The proposition can then be summarised as saying that convergent loops terminate in **Sets** (the converse being automatic). This not true in general, however. In a category of spaces, convergence may simply imply that iteration of the loop approaches a fixpoint, without requiring that it be attained in a finite number of steps. For example, the successor s on the free chain ω in $\mathbf{Pos}(\omega)$ is convergent though it does not terminate. More generally, we have the following proposition.

Proposition 2.5 *In* **Pos**(ω) *if* f *is a continuous loop which is increasing* ($x \leq f(x)$) *then it converges with universal invariant* $Yf : C \rightarrow \mathtt{Fix}(f)$ *defined by*

$$Yf(x) = \bigsqcup_n f^n(x)$$

Proof Clearly Yf is continuous and invariant for f and $Yf \circ \mathtt{fix}(f) = \mathtt{id}$. If $g : D \rightarrow Q$ is an invariant for f then $g \circ f^n(x) = g(x)$ and so

$$\begin{aligned} g \circ \mathtt{fix}(f) \circ Yf(x) &= g(\bigsqcup_n f^n(x)) \\ &= \bigsqcup_n g \circ f^n(x) \\ &= \bigsqcup_n g(x) = g(x) \end{aligned}$$

□

In general $Yf(x)$ is the least fixpoint of f greater than x. Hence, if $\perp \in C$ then $Yf(\perp)$ is the least fixpoint of f.

In general, the factorisation of an invariant through the universal such cannot be constructed, but when the loop is convergent then this is trivial.

Lemma 2.6 *If* $f : C \rightarrow C$ *converges to* $m : C_0 \rightarrow C$ *and* $g : C \rightarrow Q$ *is an invariant for* f *then*

$$g = (g \circ m) \circ \mathtt{inv}(f)$$

In particular, the universal invariant is the unique morphism $g : C \rightarrow C_0$ *satisfying the equations*

$$\begin{aligned} g \circ f &= g \\ g \circ m &= \mathtt{id} \end{aligned}$$

Although convergent loops were introduced to analyse while-loops in **Pos**(ω) [12] Cockett's characterisation of list objects using the termination of `tail` [5] provoked the realisation that the explicit bounds on iteration required for termination are not necessary for the understanding of primitive recursion on the natural numbers and finite lists, but rather that convergence suffices. We will return to this theme after introducing distributive categories.

3 Distributive Categories

It is possible, perhaps even desirable on first reading, to ignore the setting of the results below, and work with sets and functions instead of a general distributive category. The latter are not introduced so as to add more structure, but rather to remove it, as there are many constructions available in **Sets** and **Pos**(ω) which are unneccessary for the work at hand, and may be lost as more computational features are introduced to the semantics.

3.1 Distributive Cartesian Categories

Let $\mathcal{D}$ be a bicartesian category, i.e. have all finite cartesian products and coproducts. Notation is as follows. The identity morphism on A is denoted id_A or more commonly by A itself. The *pairing* of $f : C \rightarrow A$ and $g : C \rightarrow B$ is denoted $\langle f, g \rangle : C \rightarrow A \times B$ with projections $\pi_{A,B} : A \times B \rightarrow A$ and $\pi'_{A,B} : A \times B \rightarrow B$, symmetry $c_{A,B} : A \times B \rightarrow B \times A$ and diagonal $\delta_A = \langle \mathrm{id}_A, \mathrm{id}_A \rangle : A \rightarrow A \times A$. The associativity isomorphism is denoted $a_{A,B,C} : (A \times B) \times C \rightarrow A \times (B \times C)$ and the unit isomorphisms are $l_A : 1 \times A \rightarrow A$ and $r_A : A \times 1 \rightarrow A$. The terminal object is 1 with $!_A : A \rightarrow 1$ being the unique morphism. Given a morphism $b : 1 \rightarrow B$ then $b \circ ! : A \rightarrow B$ is a *constant morphism* and may be abbreviated to $b : A \rightarrow B$.

The *cases morphism* for $f : A \rightarrow C$ and $g : B \rightarrow C$ is denoted $[f, g] : A + B \rightarrow C$ with inclusions $\iota_{A,B} : A \rightarrow A + B$ and $\iota'_{A,B} : B \rightarrow A + B$. The *initial object* is 0 and with canonical morphisms $?_A : 0 \rightarrow A$.

The subscripts on natural transformations will usually be suppressed unless they aid comprehension.

For each triplet of objects A, B, C there are canonical morphisms (components of a natural transformation)

$$
\begin{array}{ccc}
(A \times B) + (A \times C) & \xrightarrow{[A \times \iota, A \times \iota']} & A \times (B + C) \\
0 & \longrightarrow & A \times 0
\end{array}
\tag{2}
$$

$\mathcal{D}$ satisfies the *distributive law* and is a *distributive cartesian category* if these are all isomorphisms. Then the inverses, labelled

$$
d_{A,B,C} : A \times (B + C) \rightarrow (A \times B) + (A \times C)
$$

are also natural.

While the name of this law has remained stable the concept "distributive category" has often included additional structures and assumptions. Walters [37] required countably infinite coproducts while Cockett [5] required all finite limits and stability of coproducts under pullback. In [13] distributive categories were tentatively dubbed *polynomial categories*. The terminology here agrees with that of Lawvere [27], and should become standard (see [6] for further discussion).

Examples of distributive categories include all bicartesian closed categories, e.g. $\mathbf{Pos}(\omega)$ since the left adjoint $A \times (-)$ preserves all colimits, including coproducts, that exist, but not $\mathbf{Pos}_\perp(\omega)$ since it lacks the coproducts.

Other (non-closed) examples include the category of topological spaces and continuous functions, and various of its sub-categories, such as metric spaces and distance-decreasing functions, and the category of countable sets and functions.

The products and coproducts allow us to express the polynomial functors appearing in the theory of datatypes, such as those for lists or binary trees on an object A

$$
\begin{aligned}
L &\cong 1+(A\times L)\\
T &\cong A+(T\times T)
\end{aligned}
$$

The prime import of the distributive law is that one can perform a case analysis in the presence of parameters. That is, to show that $f = g : A\times(B+C)\to D$ it suffices to show that $f\circ d^{-1} = g\circ d^{-1} : (A\times B)+(A\times C)\to D$ which can be established by pre-composing with the inclusions.

Consequently, the booleans can be represented by $\texttt{Bool} = 1+1$ whose inclusions are then labelled $\texttt{true} = \iota : 1\to\texttt{Bool}$ and $\texttt{false} = \iota' : 1\to\texttt{Bool}$. Predicates or tests are represented by morphisms into $\texttt{Bool}$ and conditionals are handled using distributivity. That is, if $b : A\to\texttt{Bool}$ and $f,g : A\to B$ are morphisms in $\mathcal{D}$ then define the conditional $\texttt{if}\ b\ \texttt{then}\ f\ \texttt{else}\ g = \mathrm{cond}(b,f,g)$ by

$$A \xrightarrow{\langle A,b\rangle} A\times\texttt{Bool} \cong A+A \xrightarrow{[f,g]} B$$

whose middle isomorphism arises from the distributive law

$$A\times\texttt{Bool} = A\times(1+1) \xrightarrow{d} (A\times 1)+(A\times 1) \xrightarrow{r\times r} A+A$$

Thus, for $x : X\to A$ we have

$$
\begin{aligned}
\mathrm{cond}(b,f,g)\circ x &= f\circ x && \text{if } b\circ x = \texttt{true}\\
\mathrm{cond}(b,f,g)\circ x &= g\circ x && \text{if } b\circ x = \texttt{false}
\end{aligned}
$$

3.2 Distributive Monoidal Categories

The results in this paper were originally proved for a distributive cartesian category. However, it quickly became clear that the main theorems also hold in many categories in which the product has weaker properties than usually demanded for a cartesian product, such as categories of partial maps and of relations, and even in categories with a mere tensor product lacking any projections and diagonals, such as the category of abelian groups, and models of linear logic. Hence the paper is written using the weaker notion of tensor product (briefly introduced below). If monoidal categories are unfamiliar then this section may be skipped and the rest of the paper read by interpreting $\otimes$ as $\times$ and I as 1.

Recall [26, 19] that a *symmetric monoidal category* $(\mathcal{V},\otimes,I,a,l,r,c)$ is given by a category $\mathcal{V}$ equipped with a binary functor $\otimes : \mathcal{V}^2\to\mathcal{V}$ called the *tensor product* which is associative, unitary (with unit object I) and symmetric, up to natural isomorphisms whose components are given by

$$
\begin{aligned}
a_{A,B,C} : (A\otimes B)\otimes C &\to A\otimes(B\otimes C)\\
l_A : I\otimes A &\to A\\
r_A : A\otimes I &\to A\\
c_{A,B} : A\otimes B &\to B\otimes A
\end{aligned}
$$

These isomorphisms are further required to satisfy some equations that guarantee coherence, i.e. that all "canonical" diagrams of natural transformations built from those above commute. In general, instances of a, l and r will be suppressed e.g. $C \otimes 0 : C \to C \otimes N$ denotes $(C \otimes 0) \circ r^{-1}$.

A *distributive monoidal category* is a symmetric monoidal category $\mathcal{D}$ which has all finite coproducts and satisfies the distributive law, namely that the natural transformations obtained from (2) by replacing $\times$ by $\otimes$ are isomorphisms with inverse $d_{A,B,C} : A \otimes (B + C) \to (A \otimes B) + (A \otimes C)$.

Symmetric monoidal closed categories with all finite coproducts are distributive, since $A \otimes (-)$ is a left adjoint, and so preserves all coproducts. In particular, the category **Ab** of abelian groups and group homomorphisms is distributive, as are the $*$-autonomous categories used to model linear logic [1, 35].

Fix a distributive monoidal category $\mathcal{D}$ for the rest of the paper. If $\mathcal{D}$ has a class of (admissible) monomorphisms [4, 11, 33] such that

- coproducts are stable under pulling back along admissible monomorphisms, and;
- if $m : A_0 \to A$ and $n : B_0 \to B$ are admissible monomorphisms then so is $m + n$, whose pullback along $\iota : A \to A + B$ (respectively, ι') is m (respectively, n).

then the category $\mathcal{D}_p$ of partial maps on $\mathcal{D}$ is a distributive monoidal category with its product and coproduct (and so its distributive law) inherited from $\mathcal{D}$. (In fact, Gordon Plotkin pointed out that if $\mathcal{D}$ is connected and $\mathcal{D}_p$ has coproducts then so does $\mathcal{D}$ and both the above conditions hold.) In particular $\mathbf{Pos}_{\perp,s}(\omega)$ the subcategory of $\mathbf{Pos}_{\perp}(\omega)$ of strict ($\perp$-preserving) functions is a distributive monoidal category.

One other important class of examples is given by the Kleisli categories for commutative monads, introduced by Kock [21], whose definition we will briefly review.

Recall [2] that a *triple* or *monad* $\mathbf{T} = (T, \eta, \mu)$ on a category $\mathcal{C}$ is given by a functor $T : \mathcal{C} \to \mathcal{C}$ and a pair of natural transformations $\eta : \mathrm{id}_{\mathcal{C}} \Rightarrow L$ and $\mu : L^2 \Rightarrow L$ that make μ *unitary* and *associative*, i.e. make the following diagrams commute.

$$
\begin{tikzcd}
TA \arrow[r, "\eta_{TA}"] \arrow[dr, "\mathrm{id}"'] & T^2 A \arrow[d, "\mu_A"] & TA \arrow[l, "T\eta_A"'] \arrow[dl, "\mathrm{id}"] \\
& TA &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3 A \arrow[r, "T\mu_A"] \arrow[d, "\mu_{TA}"'] & T^2 A \arrow[d, "\mu_A"] \\
T^2 A \arrow[r, "\mu_A"'] & TA
\end{tikzcd}
$$

A *strong* monad on a monoidal category $\mathcal{C}$ is a monad $\mathbf{T}$ on $\mathcal{C}$ equipped with a natural transformation $\tau_{A,B} : A \otimes TB \to T(A \otimes B)$ called its *strength*, which is *unitary* and *associative*, i.e.

$$\begin{tikzcd} 1\otimes TA \arrow[r, "\tau"] \arrow[dr, "l"'] & T(1\otimes A) \arrow[d, "Tl"] \\ & TA \end{tikzcd} \qquad \begin{tikzcd} A\otimes B\otimes TC \arrow[r, "A\otimes\tau"] \arrow[dr, "\tau"'] & A\otimes T(B\otimes C) \arrow[d, "\tau"] \\ & T(A\otimes B\otimes C) \end{tikzcd}$$

These have been used by Moggi [30] and Crole and Pitts [8] to model computation types which represent lifting, non-determinism, side-effects, continuations, etc.

If the monoidal category $\mathcal{C}$ is symmetric then the strength τ can be dualised to yield

$$\tau'_{A,B} = Tc \circ \tau \circ c : TA\otimes B \rightarrow T(A\otimes B)$$

Define the strong monad (T, η, μ, τ) to be *commutative* if the following diagram commutes

$$\begin{tikzcd} TA\otimes TB \arrow[r, "\tau"] \arrow[d, "\tau'"'] & T(TA\otimes B) \arrow[r, "T\tau'"] & T^2(A\otimes B) \arrow[dd, "\mu"] \\ T(A\otimes TB) \arrow[d, "T\tau"'] & & \\ T^2(A\otimes B) \arrow[rr, "\mu"'] & & T(A\otimes B) \end{tikzcd}$$

For example, the monads for lifting and non-determinism are commutative, but those for side-effects and continuations are not. It follows that if $\mathbf{T}$ is a commutative monad then its Kleisli category $\mathcal{D}_{\mathbf{T}}$ is a distributive monoidal category with its structure inherited from $\mathcal{D}$.

4 Natural Numbers Objects

4.1 Natural Numbers Candidates

A *natural numbers candidate*, or NNC, is an object N equipped with a point *zero* represented by a morphism $0 : I \rightarrow N$ and a loop $s : N \rightarrow N$ called the *successor* that make the following diagram a coproduct.

$$I \xrightarrow{0} N \xleftarrow{s} N$$

In other words $[0, s] : I + N \to N$ is an isomorphism whose inverse will be denoted $\mathsf{pop} : N \to I + N$.

From this limited position we can already define a little structure. For example, the zero test $\mathtt{isZero} : N \to \mathtt{Bool}$ is defined by

$$N \xrightarrow{\mathsf{pop}} I + N \xrightarrow{[\mathtt{true}, \mathtt{false}]} \mathtt{Bool}$$

and the *predecessor* $\mathtt{pred} : N \to N$ is defined by $\mathtt{pred} = [0, N] \circ \mathsf{pop}$. Thus $\mathtt{pred} \circ 0 = 0$ and $\mathtt{pred} \circ s = \mathtt{id}_N$. We can also define single steps of a recursion: if f is a loop on C then $\mathsf{once}(f)$ is the loop on $C \otimes N$ given by

$$C \otimes N \xrightarrow{C \otimes \mathsf{pop}} C \otimes (I + N) \xrightarrow{d} (C \otimes I) + (C \otimes N) \xrightarrow{[C \otimes 0, f \otimes N]} C \otimes N$$

Using elements this says

$$\begin{aligned} \mathsf{once}(f)(x, 0) &= (x, 0) \\ \mathsf{once}(f)(x, sn) &= (f(x), n) \end{aligned}$$

For example $\mathsf{once}(\mathtt{id}_I)$ corresponds to the predecessor on N.

Lemma 4.1 *A morphism $g : C \otimes N \to Q$ is an invariant for* $\mathsf{once}(f)$ *iff the following diagram commutes*

$$\begin{tikzcd} C \otimes N \arrow[r, "f \otimes N"] \arrow[d, "C \otimes s"'] & C \otimes N \arrow[d, "g"] \\ C \otimes N \arrow[r, "g"'] & Q \end{tikzcd}$$

Proof Since $d \circ (C \otimes \mathsf{pop})$ is an isomorphism we can reverse it and reform the desired commuting square as

$$\begin{tikzcd} (C \otimes I) + (C \otimes N) \arrow[r, "{[C \otimes 0, f \otimes N]}"] \arrow[d, "{[C \otimes 0, C \otimes s]}"'] & C \otimes N \arrow[d, "g"] \\ C \otimes N \arrow[r, "g"'] & Q \end{tikzcd}$$

which is amenable to case analysis. The left-hand case holds trivially and the right-hand case is the conclusion. $\square$

Lemma 4.2 *If $h : C \rightarrow D$ is a loop morphism from f to g then $h \otimes N$ is a loop morphism from* $\mathsf{once}(f)$ *to* $\mathsf{once}(g)$.

Proof By the previous lemma it suffices to prove

$$\begin{aligned}
\mathsf{once}(g) \circ (h \otimes N) \circ (C \otimes s) &= \mathsf{once}(g) \circ (C \otimes s) \circ (h \otimes N) \\
&= (g \otimes N) \circ (h \otimes N) \\
&= (h \otimes N) \circ (f \otimes N)
\end{aligned}$$

□

4.2 Initial Natural Numbers Objects

Consider the representation of the primitive recursive functions in a distributive cartesian category with a NNC. The zero, successor and predecessor are already represented, and the projections $N^k \rightarrow N$ and substitution are given by the product and composition structure of the category, so that it remains to represent iteration. That is, given primitive recursive functions $x' : N^k \rightarrow N$ and $f' : N \otimes N^k \otimes N \rightarrow N$ there is a primitive recursive function $h' : N^k \otimes N \rightarrow N$ determined by

$$\begin{aligned}
h'(y_1, y_2, \ldots, y_k, 0) &= x'(y_1, y_2, \ldots, y_k) \\
h'(y_1, y_2, \ldots, y_k, sn) &= f'(h'(y_1, y_2, \ldots, y_k, n), y_1, y_2, \ldots, y_k, n)
\end{aligned}$$

Translating these equations into diagrams yields

$$\begin{tikzcd}
N^k \arrow[r, "N^k \times 0"] \arrow[dr, "x'"'] & N^k \times N \arrow[d, "h'"] & N^k \times N \arrow[l, "N^k \times s"'] \arrow[d, "\langle h', N^k \times N \rangle"] \\
& N & N \times N^k \times N \arrow[l, "f'"']
\end{tikzcd}$$

That h' is the unique such function is usually established by induction. Here we will take the uniqueness of h as part of the construction.

The requirement that the target of f' be N seems a little arbitrary. If we allow that f' may take values in N^p for any p then the picture can be simplified as follows: replace f' by the loop $f = \langle f', (B \times s) \circ \pi'_{N, N^k \times N} \rangle$ on N^p where $p = k + 2$ and replace x' by $x = \langle x', N^k, 0 \rangle : N^k \rightarrow N^p$. Then $h = \langle h', N^k \times N \rangle : N^k \times N \rightarrow N^p$ makes the following diagram commute

$$\begin{tikzcd}
N^k \arrow[r, "N^k \times 0"] \arrow[dr, "x"'] & N^k \times N \arrow[d, "h"] & N^k \times N \arrow[l, "N^k \times s"'] \arrow[d, "h"] \\
& N^p & N^p \arrow[l, "f"]
\end{tikzcd}$$

which corresponds to the following pair of equations

$$\begin{aligned} h(y,0) &= x(y) \qquad (3) \\ h(y,s(n)) &= f(h(y,n)) \qquad (4) \end{aligned}$$

So far we have simply translated the usual definitions into diagrammatic form. Now we will modify the construction according to categorical principles, which in essence is the principled application of Occam's razor.

First, the internal structures of N^k and N^p are irrelevant to the construction of h, as are the natures of x and f, which can thus be replaced by arbitrary objects and morphisms. When x and f represent primitive recursive functions then so will h.

Second, and more radically, iteration makes no use of the pairing of the cartesian product and so, following Paré and Roman [32] we will replace it by a tensor product.

Definition 4.3 *An* initial natural numbers object *(or initial NNO) in a distributive category $\mathcal{D}$ is given by an object N equipped with a pair of morphisms* zero $0 : I \rightarrow N$ *and* successor $s : N \rightarrow N$ *which satisfy the following universal property. Given an object C equipped with a "point" $x : B \rightarrow C$ and a* loop *f on C then there is a unique morphism $h = \mathtt{It}(x,f) : B \otimes N \rightarrow C$ called the* iterator *of x and f making the following diagram commute*

$$\begin{tikzcd} B \arrow[r, "B\otimes 0"] \arrow[dr, "x"'] & B\otimes N \arrow[d, "h"] & B\otimes N \arrow[l, "B\otimes s"'] \arrow[d, "h"] \\ & C & C \arrow[l, "f"] \end{tikzcd} \qquad (5)$$

When $x = \mathrm{id}_C$ then $\mathtt{It}(x,f)$ may be abbreviated to $\mathtt{It}(f)$.

The categorical approach to primitive recursion is due to Lawvere [25]. There the category was assumed cartesian closed and so the parameter B was unneccessary, as our h could be captured in its curried form as a morphism from N to C^B. That this is false economy can be seen by comparing the definition of addition as $\mathtt{It}(s)$ with that obtained from an iterator $N \rightarrow N^N$.

It follows that $B \otimes N$ is an initial algebra for the functor $B + (-)$ [31, 36]. Consequently [23], we have an isomorphism

$$B \otimes N \cong B + (B \otimes N)$$

which when B equals I shows that N is indeed a natural numbers candidate.

Of course this use of a tensor product raises the question of how projections and pairing with respect to the natural numbers can be expressed. Remarkably, they can be recovered as iterators by

$$\begin{aligned}
\pi_{C,N} &= \mathtt{It}(\mathtt{id}_C) : C\otimes N \to C \\
\delta_N &= \mathtt{It}(0\otimes 0, s\otimes s) : N \to N\otimes N \\
c_{C,N} &= \mathtt{It}(0\otimes C, s\otimes C) : C\otimes N \to N\otimes C
\end{aligned}$$

If $\mathcal{D}$ is cartesian then it is easily checked that these definitions agree with the usual notions. Even in the general case, however, all the usual equations hold, as the sub-category of tensor powers of N in $\mathcal{D}$ form a cartesian sub-category of $\mathcal{D}$ [32].

For example, in **Ab** the initial natural numbers object is the free ring $\mathbf{Z}[x]$ on one generator x with $\mathtt{zero} : \mathbf{Z} \to \mathbf{Z}[x]$ given by inclusion and successor given by multiplication by x.

The techniques appropriate to initial algebras are too well known to warrant detailed explanation here, but we will establish some facts required below.

Lemma 4.4 *Let $(N, 0, s)$ be an initial NNO. If $h : C \to D$ is a loop morphism from $f : C \to C$ to $g : D \to D$ and $x : A \to B$ and $y : B \to C$ are morphisms then*

$$h \circ \mathtt{It}(y, f) \circ (x\otimes N) = \mathtt{It}(h \circ y \circ x, g) \tag{6}$$

Consequently we have

$$\begin{aligned}
h \circ \mathtt{It}(f) &= \mathtt{It}(g) \circ (h\otimes N) && (7)\\
\mathtt{It}(f) \circ (C\otimes s) &= f \circ \mathtt{It}(f) = \mathtt{It}(f) \circ (f\otimes N) && (8)\\
\mathtt{It}(f) \circ (x\otimes N) &= x \circ \pi : B\otimes N \to C \textit{ if } x \textit{ is fixed by } f && (9)\\
g \circ \mathtt{It}(f) &= g \circ \pi \textit{ if } g \textit{ is an invariant for } f && (10)
\end{aligned}$$

In particular (8) shows that $\mathtt{It}(f)$ *is an invariant for* $\mathtt{once}(f)$.

Proof The commutativity of

$$
\begin{tikzcd}
 & A\otimes N \arrow[d, "x\otimes N"] & A\otimes N \arrow[l, "A\otimes s"'] \arrow[d, "x\otimes N"] \\
A \arrow[ur, "A\otimes 0"] \arrow[r, "x\otimes 0"] \arrow[dr, "y\circ x"] \arrow[ddr, "h\circ y\circ x"'] & B\otimes N \arrow[d, "\mathtt{It}(y,f)"] & B\otimes N \arrow[l, "B\otimes s"'] \arrow[d, "\mathtt{It}(y,f)"] \\
 & C \arrow[d, "h"] & C \arrow[l, "f"'] \arrow[d, "h"] \\
 & D & D \arrow[l, "g"]
\end{tikzcd}
$$

shows that the left-hand side of (6) satisfies the universal property of its right-hand side.

Two applications of this result show that both sides of (7) equal $\mathtt{It}(h,g)$. This then implies the second equality of (8) upon taking $h = g = f$ while the first equation is part of the definition of $\mathtt{It}(f)$ whose invariance then follows from Lemma 4.1. For (9) observe that x is a loop morphism from $\mathtt{id}_A$ to f and apply (6) twice to see that both sides equal $\mathtt{It}(x,f)$. Similarly g is a loop morphism from f to $\mathtt{id}_Q$ in (10) and so $g\circ\mathtt{It}(f) = \pi\circ(g\otimes N) = g\circ\pi$. □

As a further corollary to (6) we have:

Lemma 4.5 *Let f, g be loops on C which satisfy $f\circ g\circ f = f\circ f\circ g$. Then $f\circ g\circ\mathtt{It}(f) = f\circ\mathtt{It}(f)\circ(g\otimes N)$.*
Proof $f\circ g$ and f are loop morphisms from f to itself. Thus

$$
\begin{aligned}
f\circ g\circ\mathtt{It}(f) &= \mathtt{It}(f)\circ(N\otimes f\circ g)\\
&= f\circ\mathtt{It}(f)\circ(N\otimes g)
\end{aligned}
$$

□

4.3 Convergent Natural Numbers Objects

Let us re-examine the equations for primitive recursion, in the case where $x = \mathtt{id}_C$ in (3). Intuitively (4) asserts that $h(y, s(n))$ is obtained by constructing the n-fold iterate of f and then applying f once more. The tail recursive approach, however, would first apply f and then iterate n times, that is, replace (4) by

$$h(y, s(n)) = h(f(y), n)$$

or equivalently

$$h(n, y) = h \circ \mathsf{once}(f)(n, y)) \tag{11}$$

which asserts that h is an invariant for the loop $\mathsf{once}(f)$ on $C \otimes N$. Now, just as the initial natural numbers were defined by means of a property universal for all the objects of the category, the natural interpretation of (11) is that h be the *universal* invariant for $\mathsf{once}(f)$, which is thus convergent by (3).

Definition 4.6 *A NNC $(N, 0, s)$ is a* convergent *natural numbers object if for every loop f on some object C the loop* $\mathsf{once}(f)$ *converges to $C \otimes 0 : C \rightarrow C \otimes N$. The universal invariant is called the* (convergent) iterator $\mathtt{Ti}(f) : C \otimes N \rightarrow C$ *of f.*

Now we will establish some basic results. The following lemma shows that convergence interacts well with parameters.

Lemma 4.7 *If f is a loop on C then* $\mathtt{Ti}(f \otimes D) = \mathtt{Ti}(f) \otimes D$ *for any object D.*
Proof Apply Lemma 2.6. □

Lemma 4.8 *If $h : C \rightarrow D$ is a loop morphism from f to g then the following diagram commutes*

$$\begin{tikzcd}
C \otimes N \arrow[r, "\mathtt{Ti}(f)"] \arrow[d, "h \otimes N"'] & C \arrow[d, "h"] \\
D \otimes N \arrow[r, "\mathtt{Ti}(g)"'] & D
\end{tikzcd}$$

Proof Lemma 4.2 shows that $h \otimes N$ is a loop morphism from $\mathsf{once}(f)$ to $\mathsf{once}(g)$ so that there is a unique morphism $C \rightarrow D$ which when substituted for the rightmost copy of h in the diagram above makes it commute, namely $\mathtt{Ti}(g) \circ (h \otimes N) \circ (C \otimes 0) = h$. □

Lemma 4.9 *If f is a loop on C in a distributive category with convergent NNO then*

$$f \circ \mathtt{Ti}(f) = \mathtt{Ti}(f) \circ (f \otimes N) = \mathtt{Ti}(f) \circ (C \otimes s) : C \otimes N \rightarrow C$$

Proof Clearly f is a loop morphism from f to f and so the previous lemma yields the first equation; the second holds by definition. □

Thus $\mathtt{Ti}(f)$ satisfies the central property of $\mathtt{It}(f)$ of being a loop morphism from $B \otimes s$ to f. We will now develop a little arithmetic in the convergent style. Define addition by $\mathtt{plus} = \mathtt{Ti}(s) : N \otimes N \rightarrow N$.

Theorem 4.10 *If f is a loop on an object C in a category with a convergent NNO then the following diagram commutes*

$$\begin{tikzcd}
C\otimes N\otimes N \arrow[r, "C\otimes\mathtt{plus}"] \arrow[d, "\mathtt{Ti}(f)\otimes N"'] & C\otimes N \arrow[d, "\mathtt{Ti}(f)"] \\
C\otimes N \arrow[r, "\mathtt{Ti}(f)"'] & C
\end{tikzcd}$$

Proof Apply Lemma 4.8 to the loop morphism $\mathtt{Ti}(f)$ from $C\otimes s$ to f. □

Theorem 4.11 *Addition is associative, unitary and commutative.*
Proof Associativity is obtained by taking $f = s : N \to N$ in the previous theorem and one unitary law follows by definition.

A straightforward application of Lemma 4.1 shows that $\mathtt{plus} \circ c : N\otimes N \to N$ is an invariant for $\mathtt{once}(s)$. Hence

$$\begin{tikzcd}
N\otimes N \arrow[r, "\mathtt{plus}"] \arrow[d, "c"'] & N \arrow[d, "h"] \\
N\otimes N \arrow[r, "\mathtt{plus}"'] & N
\end{tikzcd} \tag{12}$$

commutes for some $h : N \to N$ namely $\mathtt{plus} \circ c \circ (N\otimes 0) = \mathtt{plus} \circ (0\otimes N)$. Using elements, this says $h(x) = 0 + x$. Thus the remaining unitary law amounts to showing that $h = \mathtt{id}_N$ and then (12) will show the commutativity of addition.

Observe that stacking two copies of (12) one above the other shows that $h \circ h$ fixes the epimorphism $\mathtt{plus}$ and so is the identity. Consider the following commutative diagram.

$$\begin{tikzcd}
N \arrow[dr, "(0\otimes 0\otimes \mathrm{N})"] \arrow[drr, "0\otimes N"] \arrow[ddr, "0\otimes h"'] & & \\
& N\otimes N\otimes N \arrow[r, "\mathtt{plus}\otimes N"'] \arrow[d, "N\otimes\mathtt{plus}"] & N\otimes N \arrow[d, "\mathtt{plus}"] \\
& N\otimes N \arrow[r, "\mathtt{plus}"'] & N
\end{tikzcd}$$

Its lower edge is $h \circ h = \mathtt{id}_N$ while its upper edge is h. Again using elements this says $x = h(h(x)) = 0 + (0 + x)) = (0 + 0) + x = 0 + x = h(x)$. □

Some other arithmetic expressions can be similarly defined, for example,

- multiplication is $\pi \circ \mathtt{Ti}((\mathtt{plus} \otimes \pi') \circ \delta) \circ (0 \otimes N)$.
- truncated subtraction is $\mathtt{Ti}(\mathtt{pred})$.
- the minimum of two naturals is $\pi \circ \mathtt{Ti}((\mathtt{plus} \otimes N) \circ (N \otimes \delta)) \circ (0 \otimes N \otimes N)$.
- less-than-or-equals is given by $\mathtt{isZero} \circ \mathtt{subt} : N \otimes N \to \mathtt{Bool}$.

4.4 The Equivalence of Natural Numbers Objects

Theorem 4.12 *Let $\mathcal{D}$ be a distributive category. Then a NNC $(N, 0, s)$ is initial iff it is convergent. For every loop f on C and $x : B \to C$ we have*

$$\begin{aligned} \mathtt{Ti}(f) &= \mathtt{It}(f) & (13) \\ \mathtt{It}(x, f) &= \mathtt{Ti}(f) \circ (x \otimes N) & (14) \end{aligned}$$

Proof Assume that $(N, 0, s)$ is convergent. If $h : B \otimes N \to C$ makes (5) commute then it is a loop morphism from $B \otimes s$ to f. Thus, by Lemma 4.8 the square in the following diagram commutes.

$$\begin{tikzcd} B \otimes N \arrow[r, "B \otimes 0 \otimes N"] \arrow[dr, "x \otimes N"'] & B \otimes N \otimes N \arrow[r, "\mathtt{Ti}(B \otimes s)"] \arrow[d, "h \otimes N"] & B \otimes N \arrow[d, "h"] \\ & C \otimes N \arrow[r, "\mathtt{Ti}(f)"'] & C \end{tikzcd}$$

The triangle commutes by assumption, and the upper composite is $(B \otimes \mathtt{Ti}(s)) \circ (B \otimes 0 \otimes N) = \mathtt{id}_{B \otimes N}$ by Lemma 4.7 and the unitary law for addition. Thus $h = \mathtt{Ti}(f) \circ (x \otimes N)$, which establishes the uniqueness of the iterator. The adequacy of this choice follows directly from Lemma 4.9.

Conversely, assume that $(N, 0, s)$ is initial. Then $\mathtt{It}(f)$ is an invariant for $\mathrm{once}(f)$ by Lemmas 4.4 and $\mathtt{It}(f) \circ (C \otimes 0) = \mathtt{id}_C$ by definition.

It remains to prove that any other invariant $g : C \otimes N \to Q$ for $\mathrm{once}(f)$ factors through $\mathtt{It}(f)$. Such a g is exactly a loop morphism from $\mathrm{once}(f)$ to $\mathtt{id}_Q$ and so the result follows from Lemma 4.4.

$$\begin{aligned} g &= \pi \circ (g \otimes N) \circ (C \otimes \delta_N) \\ &= g \circ \mathtt{It}(\mathrm{once}(f) \circ (C \otimes \delta_N) \\ &= g \circ \mathtt{It}(C \otimes 0, f \otimes N) \\ &= g \circ (C \otimes 0) \circ \mathtt{It}(f) \end{aligned}$$

The third equation above is established by direct verifiction that $\mathtt{It}(\mathsf{once}(f) \circ (C\otimes\delta_N)$ satisfies the desired universal property while the fourth holds since $C\otimes 0$ is a loop morphism from f to $f\otimes N$. □

As the primitive recursive functions can all be represented using an initial NNO [24, Corollary III.2.6] this implies

Corollary 4.13 *The primitive recursive functions can be represented by a convergent NNO in a distributive category.*

5 Lists

The theory of lists developed here has many features in common with that of the natural numbers (which are, after all, simply lists on I). The primary difference is that the reverse operation, which is the identity on the NNO, now plays an important role. Also, the length of a list is used in Section 7 to provide a third characterisation of lists.

5.1 List Candidates

Let A be an object in $\mathcal{C}$. A *list candidate* for A is a solution of the domain equation for lists, i.e. is given by an object L equipped with morphisms $\mathtt{nil} : I \rightarrow L$ and $\mathsf{cons} : A\otimes L \rightarrow L$ such that $[\mathtt{nil}, \mathsf{cons}]$ is an isomorphism

$$L \cong I + (A\otimes L)$$

The inverse morphism is denoted $\mathsf{pop} : L \rightarrow I + (A\otimes L)$. A list candidate for I is just a NNC with $(N, 0, s) = (L, \mathtt{nil}, \mathsf{cons} \circ l^{-1})$. Singleton lists are created by

$$\eta = \mathsf{cons} \circ (A\otimes\mathtt{nil}) : A \rightarrow A\otimes L \rightarrow L$$

In a cartesian setting we can also define operations which lose information, such as

$$\begin{aligned} \mathtt{head} &= (1+\pi) \circ \mathsf{pop} : L \rightarrow 1 + A \\ \mathtt{tail} &= [\mathtt{nil}, \pi'] \circ \mathsf{pop} : L \rightarrow L \end{aligned}$$

A *right A-action* on an object C is a morphism $\alpha : C\otimes A \rightarrow C$. Each such has a corresponding *left* A-action given by $\alpha_c = \alpha \circ c : A\otimes C \rightarrow C$. Fix, for the rest of the paper, an object A with a list candidate $(L, \mathtt{nil}, \mathsf{cons})$ and right A-action α on an object C.

By analogy with $\mathsf{once}(f)$ we can define the loop $\mathsf{shunt}(\alpha)$ on $C\otimes L$ by

$$C\otimes L \xrightarrow{d \circ (C\otimes\mathsf{pop})} (C\otimes I) + (C\otimes A\otimes L) \xrightarrow{[C\otimes\mathtt{nil}, \alpha\otimes L]} C\otimes L$$

If the list is nil then shunt(α) is fixed. Otherwise it returns the result of making the head of the list act on C paired with the tail of the list. If L consists of finite lists then iterating shunt(α) will ultimately yield values whose list component is nil, i.e. which are fixed by shunt(α), in which case shunt(α) is convergent. The next four lemmas mimic the corresponding results for the convergent natural numbers, and are given without proof.

Lemma 5.1 *$g : C\otimes L \to Q$ is an invariant for* shunt(α) *iff the following diagram commutes*

$$\begin{tikzcd} C\otimes A\otimes L \arrow[r, "\alpha\otimes L"] \arrow[d, "C\otimes\mathtt{cons}"'] & C\otimes L \arrow[d, "g"] \\ C\otimes L \arrow[r, "g"'] & Q \end{tikzcd}$$

Lemma 5.2 *If $\beta : D\otimes A \to D$ is another right A-action and $h : C \to D$ is an A-action homomorphism (i.e. $\beta \circ (h\otimes A) = h \circ \alpha$) then $h\otimes L$ is a loop morphism from* shunt(α) *to* shunt(β).

5.2 Convergent List Objects

The list candidate $(L, \mathtt{nil}, \mathtt{cons})$ is *convergent* if, for every right A-action α on an object C, the loop shunt(α) converges to $C\otimes\mathtt{nil} : C \to C\otimes L$. The universal invariant is denoted foldl(α) and called *foldleft* of α.

In **Sets** and **Pos**(ω) this is the usual operation *foldleft* on lists. If $x \in C$ and $\alpha(x, a)$ is denoted $x\oplus a$ then

$$\mathtt{foldl}(x, [a_1, a_2, \ldots, a_n]) = (\ldots((x\oplus a_1)\oplus a_2)\ldots)\oplus a_n$$

The convergent list object on an abelian group in **Ab** is its group ring.

Lemma 5.3 *For any object D it follows that* $\mathtt{foldl}(D\otimes\alpha) = D\otimes\mathtt{foldl}(\alpha)$.

Lemma 5.4 *If $h : C \to D$ is a right A-action morphism from α to $\beta : D\otimes A \to D$ then the following diagram commutes*

$$\begin{tikzcd} C\otimes L \arrow[r, "\mathtt{foldl}(\alpha)"] \arrow[d, "h\otimes L"'] & C \arrow[d, "h"] \\ D\otimes L \arrow[r, "\mathtt{foldl}(\beta)"'] & D \end{tikzcd}$$

Lemma 5.5 $\mathtt{foldl}(\mathtt{cons}_c) \circ c : L \otimes L \to L$ *is an invariant for* $\mathtt{foldl}(\mathtt{cons}_c)$. *Thus there is a unique morphism* $h : L \to L$ *making the following diagram commute.*

$$\begin{tikzcd} L^2 \arrow[r, "\mathtt{foldl}(\mathtt{cons}_c)"] \arrow[d, "c"'] & L \arrow[d, "h"] \\ L^2 \arrow[r, "\mathtt{foldl}(\mathtt{cons}_c)"'] & L \end{tikzcd} \tag{15}$$

It is an involution called reverse *and defined by* $\mathtt{rev} = \mathtt{foldl}(\mathtt{cons}_c) \circ (\mathtt{nil} \otimes L)$. *Hence* $\mathtt{rev} \circ \mathtt{nil} = \mathtt{nil}$ *and* $\mathtt{rev} \circ \eta = \eta$.

Proof The proof of invariance is a small diagram-chase (or four line proof). Stacking two copies of (15) shows that $\mathtt{rev} \circ \mathtt{rev} = \mathtt{id}_L$ by the universal property of foldleft and the equation for $\mathtt{rev}$ follows in the usual way. The rest follows from

$$\mathtt{rev} \circ \mathtt{nil} = \mathtt{foldl}(\mathtt{cons}_c) \circ (\mathtt{nil} \otimes \mathtt{nil}) = \mathtt{nil}$$

and

$$\begin{aligned} \mathtt{rev} \circ \eta &= \mathtt{foldl}(\mathtt{cons}_c) \circ (\mathtt{nil} \otimes L) \circ \mathtt{cons} \circ (A \otimes \mathtt{nil}) \\ &= \mathtt{foldl}(\mathtt{cons}_c) \circ (L \otimes \mathtt{cons}) \circ (\mathtt{nil} \otimes A \otimes \mathtt{nil}) \\ &= \mathtt{foldl}(\mathtt{cons}_c) \circ (\mathtt{cons}_c \otimes L) \circ (\mathtt{nil} \otimes A \otimes \mathtt{nil}) \\ &= \mathtt{foldl}(\mathtt{cons}_c) \circ (L \otimes \mathtt{nil}) \circ \eta \\ &= \eta \end{aligned}$$

□

Now define $\mathtt{snoc} : L \otimes A \to L$ and $@ : L \otimes L \to L$ by

$$\begin{aligned} \mathtt{snoc} &= \mathtt{rev} \circ \mathtt{cons} \circ (A \otimes \mathtt{rev}) \circ c : L \otimes A \to L \\ @ &= \mathtt{foldl}(\mathtt{snoc}) : L \otimes L \to L \end{aligned}$$

$\mathtt{snoc}$ is like $\mathtt{cons}$ but attaches entries to the tail of the list, and @ is the append operation. Note that if $A = I$ so that $L = N$ is a convergent NNO then from the proof of Proposition 4.11 we have $\mathtt{rev} = h = \mathtt{id}_N$ which implies that $\mathtt{snoc} = s \circ r$ and $@ = \mathtt{plus}$ as expected.

Lemma 5.6 (i) $@ = \mathtt{fold}(\mathtt{cons}_c) \circ (L \otimes \mathtt{rev}) \circ c$.
(ii) $@ \circ (\mathtt{nil} \otimes L) = \mathtt{id}$.
(iii) $@ \circ (\eta \otimes L) = \mathtt{cons}$.

Proof The right-hand side of (i) is an invariant for $\mathtt{shunt(snoc)}$ since the following diagram commutes

$$\begin{tikzcd}
L\otimes A\otimes L \arrow[rr, "L\otimes c"] \arrow[dr, "c_{C\otimes A,L}"] \arrow[ddd, "\mathtt{snoc}\otimes L"'] & & L\otimes L\otimes A \arrow[r, "L\otimes\mathtt{cons}_c"] \arrow[d, "c_{L,L\otimes A}"] & L\otimes L \arrow[d, "c"] \\
& L\otimes L\otimes A \arrow[r, "c"] \arrow[dd, "L\otimes\mathtt{snoc}"] & L\otimes A\otimes L \arrow[r, "\mathtt{cons}_c\otimes L"] \arrow[d, "L\otimes A\otimes\mathtt{rev}"'] & L\otimes L \arrow[d, "L\otimes\mathtt{rev}"'] \\
& & L\otimes A\otimes L \arrow[r, "\mathtt{cons}_c\otimes L"] \arrow[d, "L\otimes\mathtt{cons}"'] & L\otimes L \arrow[d, "\mathtt{foldl(cons}_c\mathtt{)}"'] \\
L\otimes L \arrow[r, "c"'] & L\otimes L \arrow[r, "L\otimes\mathtt{rev}"'] & L\otimes L \arrow[r, "\mathtt{foldl(cons}_c\mathtt{)}"'] & L
\end{tikzcd}$$

and further

$$\begin{aligned}
\mathtt{foldl(cons}_c\mathtt{)}\circ(L\otimes\mathtt{rev})\circ c\circ(L\otimes\mathtt{nil}) &= \mathtt{foldl(cons}_c\mathtt{)}\circ(\mathtt{nil}\otimes\mathtt{rev}) \\
&= \mathtt{rev}\circ\mathtt{rev} = \mathtt{id}_L
\end{aligned}$$

Thus $\mathtt{foldl(cons}_c\mathtt{)}\circ(L\otimes\mathtt{rev})\circ c = \mathtt{foldl(snoc)} = @$ by Lemma 2.6. This and Lemma 5.5 then imply (ii) since

$$\begin{aligned}
@\circ(\mathtt{nil}\otimes L) &= \mathtt{foldl(cons}_c\mathtt{)}\circ(L\otimes\mathtt{rev})\circ c(\mathtt{nil}\otimes L) \\
&= \mathtt{foldl(cons}_c\mathtt{)}\circ(L\otimes\mathtt{nil}) = \mathtt{id}_L
\end{aligned}$$

The proof of (iii) is similar. □

Theorem 5.7 *Append is an associative and unitary operation.*
Proof One unitary law is part of the definition, the other is Lemma 5.6(ii). By the associativity of @ is meant the commutativity of

$$\begin{tikzcd}
L\otimes L\otimes L \arrow[r, "L\otimes @"] \arrow[d, "@\otimes L"'] & L\otimes L \arrow[d, "@"] \\
L\otimes L \arrow[r, "@"'] & L
\end{tikzcd}$$

First we will establish that both sides of the square are colimits for the diagram with one object $L\otimes L\otimes L$ and the two loops $\mathtt{shunt(snoc)}\otimes L$ and $L\otimes\mathtt{shunt(snoc)}$. Separately these have colimits $@\otimes L$ and $L\otimes @$ respectively which have a common splitting (right inverse) $L\otimes\mathtt{nil}\otimes L$. Thus if $g : L\otimes L\otimes L \rightarrow Q$ is invariant for both loops then it factors through each of these colimits by the same map $h = g \circ (L\otimes\mathtt{nil}\otimes L)$. Hence by Lemma 5.6 and the definition of @ we have

$$h \circ (\mathtt{snoc}\otimes L) = h \circ (@\otimes L) \circ (L\otimes\eta\otimes L) = h \circ (L\otimes @) \circ (L\otimes\eta\otimes L) = h \circ (L\otimes\mathtt{cons})$$

which shows that h is an invariant for $\mathtt{shunt(snoc)}$ and so factors through @ as required. The factorisation is unique since both $@ \circ (@\otimes L)$ and $@ \circ (L\otimes @)$ are epimorphisms, and as both colimits yield the same mediating morphism they must be equal. $\square$

Corollary 5.8 *The following equations hold.*

$$\begin{aligned} \mathtt{cons} \circ (A\otimes\mathtt{snoc}) &= \mathtt{snoc} \circ (\mathtt{cons}\otimes A) : && (16)\\ \mathtt{cons} \circ (A\otimes @) &= @ \circ (\mathtt{cons}\otimes L) && (17)\\ \mathtt{foldl}(\alpha) \circ (C\otimes\mathtt{snoc}) &= \alpha \circ (\mathtt{foldl}(\alpha)\otimes A) && (18) \end{aligned}$$

Proof The associativity of append and Lemma 5.6 imply

$$\begin{aligned} \mathtt{cons} \circ (A\otimes\mathtt{snoc}) &= @ \circ (L\otimes @) \circ (\eta\otimes L\otimes\eta)\\ &= @ \circ (@\otimes L) \circ (\eta\otimes L\otimes\eta)\\ &= \mathtt{snoc} \circ (\mathtt{cons}\otimes A) \end{aligned}$$

Thus $\mathtt{cons}$ is a right A-action homomorphism from $A\otimes\mathtt{snoc}$ to $\mathtt{snoc}$ so that Lemma 5.4 implies (17). The final equation follows upon showing that the left-hand side is an invariant for $\mathtt{shunt}(\alpha)$.

$$\begin{tikzcd}
C\otimes A\otimes L\otimes A \arrow[rr, "C\otimes\mathtt{cons}\otimes A"] \arrow[d, "\alpha\otimes L\otimes A"'] \arrow[dr, "C\otimes A\otimes\mathtt{snoc}"] & & C\otimes L\otimes A \arrow[dr, "C\otimes\mathtt{snoc}"] & \\
C\otimes L\otimes A \arrow[dr, "C\otimes\mathtt{snoc}"'] & C\otimes A\otimes L \arrow[rr, "C\otimes\mathtt{cons}"] \arrow[d, "\alpha\otimes L"] & & C\otimes L \arrow[d, "\mathtt{foldl}(\alpha)"] \\
 & C\otimes L \arrow[rr, "\mathtt{foldl}(\alpha)"'] & & C
\end{tikzcd}$$

$\square$

The proof of associativity of append can be generalised as follows. Given a morphism $u : B \to A$ define a right B-action $\mathtt{act}(u)$ on L by

$$L \otimes B \xrightarrow{(L \otimes u)} L \otimes A \xrightarrow{\mathtt{snoc}} L \tag{19}$$

Theorem 5.9 *If B has a convergent list object $(L', \mathtt{nil}', \mathtt{cons}')$ then the following diagram commutes.*

$$\begin{tikzcd}
C \otimes L \otimes L' \arrow[r, "C \otimes \mathtt{foldl}(\mathtt{act}(u))"] \arrow[d, "\mathtt{foldl}(\alpha) \otimes L'"'] & C \otimes L \arrow[d, "\mathtt{foldl}(\alpha)"] \\
C \otimes L' \arrow[r, "\mathtt{foldl}(\alpha \circ (C \otimes u))"'] & C
\end{tikzcd} \tag{20}$$

Proof It suffices to prove that $\mathtt{foldl}(\alpha)$ is a B-action morphism from $C \otimes \mathtt{act}(u)$ to $\alpha \circ (C \otimes u)$

$$\begin{tikzcd}
C \otimes L \otimes B \arrow[r, "C \otimes L \otimes u"] \arrow[d, "\mathtt{foldl}(\alpha) \otimes B"'] & C \otimes L \otimes A \arrow[r, "C \otimes \mathtt{snoc}"] \arrow[d, "\mathtt{foldl}(\alpha) \otimes A"'] & C \otimes L \arrow[d, "\mathtt{foldl}(\alpha)"'] \\
C \otimes B \arrow[r, "C \otimes u"'] & C \otimes A \arrow[r, "\alpha"'] & C
\end{tikzcd}$$

The right-hand square commutes by (18). □

Corollary 5.10 *The following diagrams commute.*

$$\begin{tikzcd}
C \otimes L \otimes L \arrow[r, "C \otimes @"] \arrow[d, "\mathtt{foldl}(\alpha) \otimes L"'] & C \otimes L \arrow[d, "\mathtt{foldl}(\alpha)"] \\
C \otimes L \arrow[r, "\mathtt{foldl}(\alpha)"'] & C
\end{tikzcd} \tag{21}$$

$$\begin{tikzcd}
C \otimes L \otimes L^2 \arrow[r, "C \otimes \mathtt{foldl}(@)"] \arrow[d, "\mathtt{foldl}(\alpha) \otimes L^2"'] & C \otimes L \arrow[d, "\mathtt{foldl}(\alpha)"] \\
C \otimes L^2 \arrow[r, "\mathtt{foldl}(\mathtt{foldl}(\alpha))"'] & C
\end{tikzcd} \tag{22}$$

Proof For the first set $u = \mathtt{id}_A$ in the theorem. Hence $\mathtt{foldl}(\alpha)$ is a loop morphism from $C \otimes @$ to $\mathtt{foldl}(\alpha)$ which yields the commutativity of the second diagram. $\square$

5.3 Initial List Objects

Finite lists are usually defined to be initial algebras, generalising the definition of the initial natural numbers.

Definition 5.11 *A list candidate* $(L, \mathtt{nil}, \mathtt{cons})$ *for* A *is a initial list object if for every left* A*-action* $\alpha' : A \otimes C \to C$ *on some object* C *and point* $x : B \to C$ *there is a unique morphism* $h = \mathtt{foldr}(x, \alpha') : L \otimes B \to C$ *called* foldright *of* x *and* α' *making the following diagram commute*

$$
\begin{tikzcd}
B \arrow[r, "\mathtt{nil} \otimes B"] \arrow[dr, "x"'] & L \otimes B \arrow[d, "h"] & A \otimes L \otimes B \arrow[l, "\mathtt{cons} \otimes B"'] \arrow[d, "A \otimes h"] \\
 & C & A \otimes C \arrow[l, "\alpha'"]
\end{tikzcd}
\tag{23}
$$

If $x = \mathtt{id}_C$ *then* $\mathtt{foldr}(x, \alpha')$ *may be abbreviated to* $\mathtt{foldr}(\alpha')$.

In **Sets** this is the usual operation *foldright* on lists. Thus if $x : 1 \to C$ is an element of C and $\alpha'(a, x)$ is denoted $a \oplus x$ then

$$\mathtt{foldr}([a_1, a_2 \ldots, a_n], x) = a_1 \oplus (a_2 \oplus (\ldots (a_n \oplus x) \ldots))$$

For example, we have

$$
\begin{aligned}
@' &= \mathtt{foldr}(L, \mathtt{cons}) : L \otimes L \to L \\
\mathtt{snoc}' &= @' \circ (L \otimes \eta) : L \otimes A \to L \\
\mathtt{rev}' &= \mathtt{foldr}(\mathtt{nil}, \mathtt{snoc}'_c) \circ r^{-1} : L \to L
\end{aligned}
$$

The primes are used to temporarily distinguish these operations from the convergent concepts introduced above.

Lemma 5.12 *The following equations hold (where* α' *is a left* A*-action on* C*).*

$$
\begin{aligned}
\mathtt{snoc}' \circ (\mathtt{cons} \otimes A) &= \mathtt{cons} \circ (A \otimes \mathtt{snoc}') && (24) \\
\mathtt{rev}' \circ \mathtt{snoc}' &= \mathtt{cons} \circ (A \otimes \mathtt{rev}') \circ c : L \otimes A \to L && (25) \\
\mathtt{rev}' \circ \mathtt{rev}' &= \mathtt{id} : L \to L && (26) \\
\mathtt{foldr}(\alpha') \circ (L \otimes \alpha') &= \mathtt{foldr}(\alpha') \circ (\mathtt{snoc} \otimes C) && (27)
\end{aligned}
$$

Proof For (24) we have:

$$\begin{aligned}
\mathtt{snoc}' \circ (\mathtt{cons}\otimes A) &= @' \circ (L\otimes\eta) \circ (\mathtt{cons}\otimes A) \\
&= @' \circ (\mathtt{cons}\otimes L) \circ (A\otimes L\otimes\eta) \\
&= \mathtt{cons} \circ (A\otimes @') \circ (A\otimes L\otimes\eta) \\
&= \mathtt{cons} \circ (A\otimes \mathtt{snoc}')
\end{aligned}$$

Both sides of (25) equal $\mathtt{foldr}(\eta, \mathtt{snoc}')$. Now use this to show that $\mathtt{rev}' \circ \mathtt{rev}' = \mathtt{foldr}(\mathtt{nil}, \mathtt{cons}) = \mathtt{id}_L$. Finally, use (24) to show that both sides of (27) equal $\mathtt{foldr}(\alpha', \alpha')$. □

The proof below that initial lists are convergent will require some information about the length of a list, which presumes that there is an (initial) natural numbers object $(N, 0, s)$. Computing the length as a morphism $L \to N$ entails loss of information, which is unreasonable. Instead we will define $b : L \to L\otimes N$ by

$$\begin{tikzcd}
I \arrow[r, "\mathtt{nil}"] \arrow[dr, "\mathtt{nil}\otimes 0"'] & L \arrow[d, "b"] & A\otimes L \arrow[l, "\mathtt{cons}"'] \arrow[d, "A\otimes b"] \\
& L\otimes N & A\otimes L\otimes N \arrow[l, "\mathtt{cons}\otimes s"]
\end{tikzcd} \tag{28}$$

It follows that $\pi \circ b = \mathtt{id}_L$. If, further, the products in $\mathcal{D}$ are cartesian then $\# = \pi' \circ b : L \to N$ is the *length* morphism for L which can then be described directly by

$$L \xrightarrow{r^{-1}} L\times I \xrightarrow{\mathtt{foldr}(0, S\circ\pi')} N \tag{29}$$

Lemma 5.13 $b \circ \mathtt{snoc}' = (\mathtt{snoc}'\otimes s) \circ (L\otimes c) \circ (b\otimes A) : L\otimes A \to L\otimes N$
Proof Both sides equal $\mathtt{It}(\eta\otimes s(0), \mathtt{cons}\otimes s)$. □

Lemma 5.14

$$\mathtt{It}(\mathtt{shunt}(\alpha)) \circ (C\otimes b) \circ (C\otimes \mathtt{rev}') \circ c = (C\otimes \mathtt{nil}) \circ \mathtt{foldr}(\alpha_c)$$

Proof It is easily established that the right-hand side equals $\mathtt{foldr}(\mathtt{shunt}(\alpha) \circ (C\otimes\mathtt{snoc}) \circ c_{A,C\otimes L})$ since $C\otimes\mathtt{nil} : C \to C\otimes L$ is a left A-action morphism from α_c to $\mathtt{shunt}(\alpha) \circ (C\otimes\mathtt{snoc}) \circ c_{A,C\otimes L}$. The nil condition for the left-hand side is straightforward. That for cons reduces to the following diagram obtained by stripping off some instances of the symmetry c (where $\mathtt{sh}$ denotes $\mathtt{shunt}(\alpha)$).

$$
\begin{tikzcd}
L\otimes C \arrow[d, "C\otimes\mathtt{rev}'"] & & & C\otimes A\otimes L \arrow[lll, "\mathtt{cons}\otimes C"'] \arrow[d, "C\otimes A\otimes\mathtt{rev}'"'] \arrow[dll, phantom, "(\mathrm{I})"] \\
C\otimes L \arrow[d, "C\otimes b"] \arrow[dr, phantom, "(\mathrm{II})"] & & C\otimes L\otimes A \arrow[ll, "C\otimes\mathtt{snoc}'"'] \arrow[dr, "C\otimes b\otimes A"] & C\otimes A\otimes L \arrow[l, "C\otimes c"'] \\
C\otimes L\otimes N \arrow[d, "\mathtt{It}(\mathtt{sh})"] \arrow[dr, phantom, "(\mathrm{III})"] & C\otimes L\otimes N \arrow[l, "C\otimes L\otimes s"'] \arrow[d, "\mathtt{It}(\mathtt{sh})"] \arrow[dr, phantom, "(\mathrm{IV})"] & C\otimes L\otimes A\otimes N \arrow[l, "C\otimes\mathtt{snoc}'\otimes N"'] \arrow[dr, "\mathtt{It}(\mathtt{sh}\otimes A)"'] & C\otimes L\otimes N\otimes A \arrow[l, "C\otimes L\otimes c"'] \arrow[d, "\mathtt{It}(\mathtt{sh})\otimes\mathrm{A}"'] \\
C\otimes L & C\otimes L \arrow[l, "\mathtt{sh}"'] & & C\otimes L\otimes A \arrow[ll, "C\otimes\mathtt{snoc}'"]
\end{tikzcd}
$$

Cell (I) commutes by (25), cell (II) is the lemma above and cell (III) is an instance of (8).

Finally we will prove that cell (IV) commutes on composition with $\mathtt{shunt}(\alpha)$ as an application of Lemma 4.5 with $f = (\mathtt{shunt}(\alpha)\otimes L)$ and $g = C\otimes\mathtt{shunt}(\mathtt{snoc})$. That $f \circ g \circ f = f \circ f \circ g$ is established by a case analysis. If both lists are constructed by $\mathtt{cons}$ then we have

$$
\begin{aligned}
g \circ f \circ (C\otimes\mathtt{cons}\otimes\mathtt{cons}) &= (\alpha\otimes\mathtt{snoc}\otimes L) \\
&= f \circ g \circ (C\otimes\mathtt{cons}\otimes\mathtt{cons})
\end{aligned}
$$

The other three cases are easy since one or more of the component morphisms vanishes. Thus $f \circ g \circ \mathtt{It}(f) = f \circ \mathtt{It}(f) \circ (N\otimes g)$. Now precomposing with $N\otimes C\otimes L\otimes\eta$ and post-composing with $C\times @'$ yields the desired result. □

5.4 The Equivalence of List Objects

Theorem 5.15 *Let $\mathcal{D}$ be a distributive category. Convergent list objects are initial. Conversely, if there is an initial natural numbers object then initial list objects are convergent. The fold operations are related by*

$$
\begin{aligned}
\mathtt{foldl}(\alpha) &= \mathtt{foldr}(C, \alpha_c) \circ (\mathtt{rev}'\otimes C) \circ c : C\otimes L \to C \\
\mathtt{foldr}(x, \alpha_c) &= \mathtt{foldl}(\alpha) \circ (x\otimes\mathtt{rev}) \circ c : L\otimes B \to C
\end{aligned}
$$

Proof In fact, when the tensor product is cartesian then the existence of a NNO is not required for the converse. Full details can be found in [14].

Assume that $(L, \mathtt{nil}, \mathtt{cons})$ is an initial list object. The following diagram shows that $\mathtt{foldr}(\alpha_c) \circ (\mathtt{rev}' \otimes C) \circ c$ is an invariant for $\mathtt{shunt}(\alpha)$.

$$\begin{tikzcd}
C\otimes A\otimes L \arrow[rrr, "\alpha\otimes L"] \arrow[dr, "c_{C\otimes A,L}"] \arrow[ddr, "c_{C,A\otimes L}"'] \arrow[ddd, "C\otimes\mathtt{cons}"'] & & & C\otimes L \arrow[d, "c"] \\
& L\otimes C\otimes A \arrow[rr, "L\otimes\alpha"] \arrow[d, "c_{L\otimes C,A}"] & & L\otimes C \arrow[d, "\mathtt{rev}'\otimes C"'] \\
& A\otimes L\otimes C \arrow[r, "(\mathtt{rev}'\otimes\mathtt{id})\circ(c\otimes C)"] \arrow[d, "\mathtt{cons}\otimes C"] & L\otimes A\otimes C \arrow[r, "L\otimes\alpha_c"] \arrow[d, "\mathtt{snoc}'\otimes C"'] & L\otimes C \arrow[d, "\mathtt{foldr}(\alpha_c)"'] \\
C\otimes L \arrow[r, "c"] & L\otimes C \arrow[r, "\mathtt{rev}'\otimes C"] & L\otimes C \arrow[r, "\mathtt{foldr}(\alpha_c)"] & C
\end{tikzcd}$$

The square in the lower right commutes by (27). The convergence equation is shown by

$$foldr(\alpha_c) \circ (\mathtt{rev}' \otimes C) \circ c \circ (C \otimes \mathtt{nil}) = \mathtt{foldr}(\alpha_c) \circ (\mathtt{nil} \otimes C) = \mathtt{id}_C$$

Now consider an arbitrary invariant $g : C\otimes L \rightarrow Q$ for $\mathtt{shunt}(\alpha)$ and let $(N, 0, s)$ be the initial NNO. The following diagram commutes

$$\begin{tikzcd}
C\otimes L\otimes N \arrow[r, "\mathtt{It}(\mathtt{shunt}(\alpha))"] \arrow[d, "g\otimes N"'] & C\otimes L \arrow[d, "g"] \\
Q\otimes N \arrow[r, "\pi"'] & Q
\end{tikzcd}$$

since both sides equal $\mathtt{It}(g, Q)$. Thus we have

$$\begin{aligned}
g &= g \circ (C\otimes\pi) \circ (C\otimes b) \\
&= \pi \circ (g\otimes N) \circ (C\otimes b) \\
&= g \circ \mathtt{It}(\mathtt{shunt}(\alpha)) \circ (C\otimes b) \\
&= g \circ (C\otimes\mathtt{nil}) \circ \mathtt{foldr}(C, \alpha_c) \circ (\mathtt{rev}' \times C) \circ c
\end{aligned}$$

by Lemma 5.14, which shows that g factors through the desired invariant.

Conversely, let $(L, \mathtt{nil}, \mathtt{cons})$ be a convergent list object for A, and let $h : L \otimes B \rightarrow C$ be a candidate for the iterator of $x : B \rightarrow C$ and $\alpha' = \alpha_c$ i.e. make (23) commute. Then h is a right A-action homomorphism from $(\mathtt{cons}_A \otimes B) \circ c_{L \otimes B, A}$ to α and so the lower rectangle of the following diagram (in which $k = \mathtt{foldl}(\mathtt{cons}_c) \otimes B$ commutes.

$$\begin{tikzcd}
 & L \otimes L \otimes B \arrow[dd, "c \otimes B"] \arrow[dr, "k"] & \\
L \otimes B \arrow[ur, "L \otimes \mathtt{nil} \otimes \mathtt{id}"] \arrow[dd, "\mathtt{nil} \otimes c"'] & & L \otimes B \arrow[dd, "\mathtt{rev} \otimes B"] \\
 & L \otimes L \otimes B \arrow[dr, "k"] & \\
L \otimes B \otimes L \arrow[ur, "L \otimes c"] \arrow[rr, "\mathtt{foldl}((\mathtt{cons} \otimes B) \circ c)"'] \arrow[d, "h \otimes L"'] & & L \otimes B \arrow[d, "h"] \\
C \otimes L \arrow[rr, "\mathtt{foldl}(\alpha)"'] & & C
\end{tikzcd}$$

The triangle commutes since $c : L \otimes B \rightarrow B \otimes L$ is a loop morphism from $(\mathtt{cons} \otimes B) \circ c$ to $\mathtt{cons}_c \otimes B$. Stripping B from the upper right square yields the definition of reverse. Now the top edge of the diagram is the identity and the left edge is $(x \otimes L) \circ c$ since $h \circ (\mathtt{nil} \otimes B) = x$. Thus inverting $\mathtt{rev}$ shows that

$$h = \mathtt{foldl}(\alpha) \circ (x \otimes L) \circ c \circ (\mathtt{rev} \otimes B) = \mathtt{foldl}(\alpha) \circ (x \otimes \mathtt{rev}) \circ c$$

which shows that this is the sole candidate for the iterator. It remains to prove that it does satisfy the conditions.

The compatibility of h with $\mathtt{nil}$ is straightforward. That for $\mathtt{cons}$ is shown by the commutativity of the following diagram whose bottom-right square commutes by (18).

$$
\begin{tikzcd}
A\otimes L\otimes B \arrow[rrr, "\mathtt{cons}\otimes B"] \arrow[d, "A\otimes c"] & & & L\otimes B \arrow[d, "c"'] \\
A\otimes B\otimes L \arrow[r, "c\otimes L"] \arrow[d, "A\otimes x\otimes \mathtt{rev}"] & B\otimes A\otimes L \arrow[rr, "B\otimes \mathtt{cons}"] \arrow[d, "x\otimes A\otimes \mathtt{rev}"] & & B\otimes L \arrow[d, "x\otimes \mathtt{rev}"'] \\
A\otimes C\otimes L \arrow[r, "c\otimes L"] \arrow[d, "A\otimes \mathtt{foldl}(\alpha)"] & C\otimes A\otimes L \arrow[r, "C\otimes c"] & C\otimes L\otimes A \arrow[r, "C\otimes \mathtt{snoc}"] \arrow[d, "\mathtt{foldl}(\alpha)\otimes A"'] & C\otimes L \arrow[d, "\mathtt{foldl}(\alpha)"'] \\
A\otimes C \arrow[rr, "c"'] & & C\otimes A \arrow[r, "\alpha"'] & C
\end{tikzcd}
$$

□

Corollary 5.16 *Corresponding operations for initial and convergent list objects are equal. That is* $@' = @$ *and* $\mathtt{snoc}' = \mathtt{snoc}$ *and* $\mathtt{rev}' = \mathtt{rev}$.

Proof Use Lemma 5.6 and Corollary 5.8 to show that

$$
\begin{aligned}
@' &= \mathtt{foldr}(L, \mathtt{cons}) = \mathtt{foldl}(\mathtt{cons}_c) \circ (L\otimes \mathtt{rev}) \circ c = @ \\
\mathtt{snoc}' &= @' \circ (L\otimes\eta) = @ \circ (L\otimes\eta) = \mathtt{snoc} \\
\mathtt{rev}' &= \mathtt{foldr}(\mathtt{nil}, \mathtt{snoc}'_c) \circ r^{-1} \\
&= \mathtt{foldl}(\mathtt{snoc}) \circ (\mathtt{nil}\otimes \mathtt{rev}) \circ c \circ r^{-1} \\
&= @ \circ (\mathtt{nil}\otimes L) \circ \mathtt{rev} = \mathtt{rev}
\end{aligned}
$$

□

5.5 Lists in Kleisli Categories

If $\mathbf{T}$ is a commutative monad on $\mathcal{D}$ then $\mathcal{D}_{\mathbf{T}}$ is a distributive monoidal category and so Theorem 5.15 applies. The existence of list objects is handled by the following theorem, whose proof will be the goal of this sub-section.

Theorem 5.17 *The free functor* $F_{\mathbf{T}} : \mathcal{D} \to \mathcal{D}_{\mathbf{T}}$ *preserves convergent list objects. Hence if* $\mathcal{D}$ *has all list objects then so does* $\mathcal{D}_{\mathbf{T}}$.

Let $(L, \mathtt{nil}, \mathtt{cons})$ be a (convergent) list object for A in $\mathcal{D}$. Clearly it is a list candidate for A in $\mathcal{D}_{\mathbf{T}}$ since the free functor preserves both the tensor and the coproduct. Now consider a right A-action $\alpha : C{\otimes}A{\to}TC$ on C in $\mathcal{D}_{\mathbf{T}}$. (All morphisms of $\mathcal{D}_{\mathbf{T}}$ will be presented as morphisms of $\mathcal{D}$.)

Define $\alpha^{\dagger} = \mu \circ T\alpha \circ \tau' : TC{\otimes}A{\to}TC$. We will see that $\mathtt{shunt}(\alpha)$ converges in $\mathcal{D}_{\mathbf{T}}$ with universal invariant $\mathtt{foldl}(\alpha^{\dagger}) \circ (\eta{\otimes}L) : C{\otimes}L{\to}TC$.

Lemma 5.18 *Let $\beta : C{\otimes}A{\to}C$ be a right A-action on C in $\mathcal{D}$. Then*

$$T\mathtt{foldl}(\beta) \circ \tau' = \mathtt{foldl}(T\beta \circ \tau')$$

Proof The following diagram shows that the left-hand side is an invariant for $\mathtt{shunt}(T\beta \circ \tau')$ and the result follows in the usual way.

$$\begin{tikzcd}
TC{\otimes}A{\otimes}L \arrow[r, "\tau'{\otimes}L"] \arrow[dd, "TC{\otimes}\mathtt{cons}"'] \arrow[dr, "\tau'"'] & T(C{\otimes}A){\otimes}L \arrow[r, "T(\beta){\otimes}L"] \arrow[d, "\tau'"] & TC{\otimes}L \arrow[d, "\tau'"] \\
 & T(C{\otimes}A{\otimes}L) \arrow[r, "T(\beta{\otimes}L)"] \arrow[d, "T(C{\otimes}\mathtt{cons})"'] & T(C{\otimes}L) \arrow[d, "T(\mathtt{foldl}(\beta))"] \\
TC{\otimes}L \arrow[r, "\tau'"'] & T(C{\otimes}L) \arrow[r, "T\mathtt{foldl}(\beta)"'] & TC
\end{tikzcd}$$

□

The commutativity of the following diagram shows that $\mathtt{foldl}(\alpha^{\dagger}) \circ (\eta{\otimes}L)$ is an invariant for $\mathtt{shunt}(\alpha)$.

$$\begin{tikzcd}
C{\otimes}A{\otimes}L \arrow[rr, "\alpha{\otimes}L"] \arrow[dr, "\eta{\otimes}\mathrm{id}"'] \arrow[ddd, "C{\otimes}\mathtt{cons}"'] & & TC{\otimes}L \arrow[r, "\tau'"] \arrow[dd, "T\eta{\otimes}L"] & T(C{\otimes}L) \arrow[d, "T(\eta{\otimes}L)"] \\
 & TC{\otimes}A{\otimes}L \arrow[u, phantom, "(\mathrm{I})"] \arrow[ur, "\alpha^{\dagger}{\otimes}L"'] \arrow[r, phantom, "(\mathrm{II})"] \arrow[ddr, "TC{\otimes}\mathtt{cons}"'] & & T(TC{\otimes}L) \arrow[d, "T\mathtt{foldl}(\alpha^{\dagger})"] \\
 & & T^2C{\otimes}L \arrow[ur, "\tau'"] \arrow[r, "\mathtt{foldl}(\alpha^{\dagger\dagger})"'] \arrow[d, "\mu{\otimes}L"] \arrow[dr, phantom, "(\mathrm{IV})"] & T^2C \arrow[d, "\mu"] \arrow[ul, phantom, "(\mathrm{III})"] \\
C{\otimes}L \arrow[rr, "\eta{\otimes}L"'] & & TC{\otimes}L \arrow[r, "\mathtt{foldl}(\alpha^{\dagger})"'] & TC
\end{tikzcd}$$

The commutativity of (I) is straightforward; (II) commutes on composition with $\texttt{foldl}(\alpha^\dagger)$ since $\mu \circ T\eta = \texttt{id}$; (III) is an instance of the lemma above, and; (IV) commutes since μ is a right A-action morphism from $\alpha^{\dagger\dagger}$ to $\alpha^\dagger$.

The convergence equation is established by

$$\texttt{foldl}(\alpha^\dagger) \circ (\eta \otimes L) \circ (C \otimes \texttt{nil}) = \texttt{foldl}(\alpha^\dagger) \circ (TC \otimes \texttt{nil}) \circ \eta = \eta$$

Now if $g : C \otimes L \to TQ$ is any other invariant for $\texttt{shunt}(\alpha)$ then routine diagram-chasing shows that $\mu \circ Tg \circ \tau'$ is an invariant for $\texttt{shunt}(\alpha^\dagger)$ and so factors through $\texttt{foldl}(\alpha^\dagger)$ whence g factors through $\texttt{foldl}(\alpha^\dagger) \circ (\eta \otimes L)$ as required.

6 Global List Properties

Assume now that every object A of $\mathcal{D}$ has a list object $(LA, \texttt{nil}_A, \texttt{cons}_A)$. It is more or less immediate from the definitions that the construction of initial lists is functorial, and in fact is a monad. For completeness these results will be developed here from the convergent list definition.

If $u : B \to A$ in $\mathcal{D}$ then define $Lu = \texttt{map}(u) : LB \to LA$ by

$$LB \xrightarrow{\texttt{nil} \otimes LB} LA \otimes LB \xrightarrow{\texttt{foldl}(\texttt{act}(u))} LA$$

where $\texttt{act}(u)$ is defined by (19).

Lemma 6.1 *Let $u : B \to A$ be a morphism. Then*

$$\texttt{foldl}(\alpha) \circ (\texttt{id} \otimes Lu) = \texttt{foldl}(\alpha \circ (C \otimes u))$$

Proof Apply Theorem 5.9. □

Theorem 6.2 *The list construction is functorial.*

Proof With u as above and $v : D \to B$ another morphism the lemma above implies

$$\begin{aligned} Lu \circ Lv &= \texttt{foldl}(\texttt{act}(u)) \circ (LA \otimes Lv) \circ (\texttt{nil}_A \otimes LD) \\ &= \texttt{foldl}(\texttt{act}(u) \circ (LA \otimes v)) \circ (\texttt{nil}_A \otimes LD) \\ &= \texttt{foldl}(\texttt{act}(u \circ v)) \circ (\texttt{nil}_A \otimes LD) \\ &= L(u \circ v) \end{aligned}$$

which shows that L preserves composition. Now $L\texttt{id}_A = @ \circ (\texttt{nil} \otimes LA) = \texttt{id}_{LA}$ completes the proof. □

Of course the generic list operations are all natural transformations. The naturality of $\texttt{nil}$ is part of the definition of L on morphisms. That of cons is the commutativity of the outer rectangle of the following diagram

$$\begin{tikzcd}
B\otimes LB \arrow[rr, "\mathtt{cons}"] \arrow[d, "B\otimes\mathtt{nil}\otimes LB"'] & & LB \arrow[d, "\mathtt{nil}\otimes LB"] \\
B\otimes LA\otimes LB \arrow[rr, shift left, "(LA\otimes\mathtt{cons})\circ(c\otimes LB)"] \arrow[rr, shift right, "(\mathtt{cons}\circ(u\otimes LA))\otimes LB"'] \arrow[d, "B\otimes\mathtt{foldl}(\mathtt{act}(u))"'] & & LA\otimes LB \arrow[d, "\mathtt{foldl}(\mathtt{act}(u))"] \\
B\otimes LA \arrow[rr, "\mathtt{cons}\circ(u\otimes LA)"'] & & LA
\end{tikzcd}$$

The lower square commutes since $\mathtt{cons}\circ(u\otimes LA)$ is a right B-action morphism from $B\otimes\mathtt{act}(u)$ to $\mathtt{act}(u)$ (a consequence of (16). The parallel morphisms are coequalised by the invariant $\mathtt{foldl}(\mathtt{act}(f))$.

Many other list constructions can be shown natural by applying the following theorem, whose proof is a routine application of invariance techniques.

Theorem 6.3 *Let $F, G : \mathcal{C}\to\mathcal{D}$ be functors. If $\gamma_A : FA\otimes GA\to FA$ is a natural right action of G on F then $\mathtt{foldl}(\gamma)_A = \mathtt{foldl}(\gamma_A)$ is natural in A.*

As a simple application we have the naturality of $\mathtt{rev} = \mathtt{foldl}(\mathtt{cons}_c)\circ(\mathtt{nil}\otimes LA)$ since pairing and composition preserve naturality. Hence snoc is natural and thus so is $\mathtt{foldl}(\mathtt{snoc}) = @$. Similarly, we can define the *flattenning* of a list of lists by the natural transformation

$$\mu_A = \mathtt{foldl}(@)\circ(\mathtt{nil}\otimes L(LA)) : L(LA)\to LA$$

Theorem 6.4 *The structure (L,η,μ) defines a monad.*
Proof To simplify the notation let $L = LA$ and L^n be the result of n applications of L to A, and let $\mu' = \mathtt{foldl}(@)$.

The proofs of the unitary laws for the monad are routine. Associativity of μ reduces, on stripping away multiple instances of nil, to showing the commutativity of

$$
\begin{tikzcd}
L\otimes L^3 \arrow[rr, "L\otimes L\mu"] \arrow[d, "L\otimes\mathtt{nil}\otimes L^3"'] \arrow[dr, "\mathtt{id}"] & & L\otimes L^2 \arrow[dd, "\mu'"] \\
L\otimes L^2\otimes L^3 \arrow[r, "\mu'\otimes L^3"'] \arrow[d, "L\otimes\mu'"'] & L\otimes L^3 \arrow[dr, "\mathtt{foldl}(\mu')"'] & \\
L\otimes L^2 \arrow[rr, "\mu'"'] & & L
\end{tikzcd}
\tag{30}
$$

The commutativity of the lower square is obtained by setting $\alpha = @$ in (22). The right-hand triangle commutes since

$$
\begin{aligned}
\mu' \circ (L\otimes L\mu) &= \mathtt{foldl}(@) \circ (L\otimes L\mu) \\
&= \mathtt{foldl}(@ \circ (L\otimes\mu)) \\
&= \mathtt{foldl}(\mu')
\end{aligned}
$$

where the second equation is an application of Lemma 6.1 and third follows since $\mu' = @ \circ (L\otimes\mu)$ follows from (22) with $\alpha = \mathtt{snoc}$. $\square$

If the product on $\mathcal{D}$ is cartesian then we can define an operation $\tau_{A,B} : A\times LB \to L(A\times B)$ sometimes called *pairwith* by

$$
A\times LB \xrightarrow{\mathtt{foldl}(\langle\mathtt{snoc},\pi_2\rangle)\circ\langle\mathtt{nil},A\times LB\rangle} L(A\times B)\times A \xrightarrow{\pi} L(A\times B)
$$

where $\pi_2 : L(A\times B)\times A\times B \to A$ is the second projection. In **Sets** the strength is given by $\tau(a,[b_i]) = [(a,b_i)]$.

Corollary 6.5 *If $\mathcal{D}$ is a distributive cartesian category then (L,η,μ,τ) is a commutative monad.*

Proof The proofs of the properties of the strength are left to the reader. Its associativity is established in the style of Theorem 5.9 but now with an additional parameter. $\square$

7 Termination

Terminating loops in **Sets** were briefly considered in Section 2. Now that we have examined natural numbers objects in abstract we can describe termination formally. This yields a third characterisation of lists in a distributive cartesian category with NNO, without assuming the various additional exactness conditions of Cockett's original result [5].

7.1 Terminating Loops

If f is a loop on C and $n : C \to N$ is a morphism then the *nth iterate of f* is $f^n : C \to C$ defined by

$$C \xrightarrow{\langle C,n\rangle} C \times N \xrightarrow{\mathtt{Ti}(f)} C$$

If f^n is fixed by f then n is a *bound* on f.

Proposition 7.1 *Let $n : C \to N$ be a bound on the loop f on C. If $p : C \to N$ is any other such then $f^n = f^p$. Hence f^n is an invariant for f.*

Proof Abbreviating $\mathtt{plus} \circ \langle n, p\rangle$ to $n + p$ we have

$$\begin{aligned} f^{n+p} &= \mathtt{Ti}(f) \circ \langle C, n+p\rangle \\ &= \mathtt{Ti}(f) \circ (\mathtt{Ti}(f) \times N) \circ \langle C, n, p\rangle \\ &= \mathtt{Ti}(f) \circ (f^n \times N) \circ \langle C, p\rangle \\ &= f^n \circ \pi \circ \langle C, p\rangle \\ &= f^n \end{aligned}$$

where the second equation holds by Theorem 4.10 and the fourth by (9). Now the commutativity of plus and the symmetry between n and p show that $f^n = f^p$. For the invariance we have

$$\begin{aligned} f^n \circ f &= \mathtt{Ti}(f) \circ (N \times f) \circ \langle n \circ f, C\rangle \\ &= \mathtt{Ti}(f) \circ (S \times C) \circ \langle n \circ f, C\rangle \\ &= f^{sonof} \end{aligned}$$

Now $f^n \circ f$ and thus f^{sonof} is fixed by f whence $f^{sonof} = f^n$ as required. □

Theorem 7.2 *Let f be a loop on C with bound $n : C \to N$. If this boundedness is witnessed by a factorisation $f^n = m \circ h$ where $m : C_0 \to C$ is a subobject of C that is fixed by f then f converges to m which is $\mathtt{fix}(f)$ and has universal invariant h.*

Proof If x is fixed by f then $x = f^n \circ x = m \circ h \circ x$ by (9) and so $m = \mathtt{fix}(f)$ is the fixpoints of f. Now taking $x = m$ shows that $m \circ h \circ m = m$ whence $h \circ m = \mathtt{id}_{C_0}$ since m is a monomorphism. Also h is an invariant for f since f^n is. Finally, if g is any invariant for f then $g \circ f^n = g \circ \pi \circ \langle C, n\rangle = g$ by (10) and so g factors through h. □

A loop f satisfying the conditions of this theorem is said to *terminate* at m with bound n or be a *terminating loop*. If f^n (for $n : C \to N$) is an invariant for f then f is a *contraction* [5]. Thus the terminating loops are contractions.

7.2 Terminating List Objects

Fix a list candidate $(L, \mathtt{nil}, \mathtt{cons})$ for A. It is a *terminating list object* if its $\mathtt{tail}$ is a terminating loop which has fixpoints given by $\mathtt{nil} : 1 \rightarrow L$.

Theorem 7.3 *Let $\mathcal{D}$ be a distributive cartesian category with a natural numbers object. Then a list candidate is initial iff terminating iff convergent.*

Proof If $(L, \mathtt{nil}, \mathtt{cons})$ is initial then we will show that

$$\mathtt{tail}^{\#} = \mathtt{foldr}(\mathtt{nil}, \pi') = \mathtt{nil}$$

(where the length $\# : L \rightarrow N$ was defined by (29)). The $\mathtt{nil}$ condition is trivial, while that for cons is the commutativity of

$$\begin{tikzcd}
L \arrow[d, "\langle L,\#\rangle"'] & & A\times L \arrow[ll, "\mathtt{cons}"'] \arrow[d, "A\times\langle L,\#\rangle"] \\
L\times N \arrow[dd, "\mathtt{It(tail)}"'] & L\times N \arrow[l, "L\times s"'] \arrow[d, "\mathtt{tail}\times N"'] & A\times L\times N \arrow[l, "\mathtt{cons}\times N"'] \arrow[dl, "\pi'\times N"] \arrow[dd, "A\times\mathtt{It(tail)}"] \\
 & L\times N \arrow[dl, "\mathtt{It(tail)}"] & \\
L & & A\times L \arrow[ll, "\pi'"]
\end{tikzcd}$$

If L is a terminating list object then its convergence will follow upon proving that $\mathtt{shunt}(\alpha)$ terminates at $\langle C, \mathtt{nil}\rangle$. If $\mathtt{tail}$ has bound $n : L \rightarrow N$ then

$$\begin{tikzcd}
C\times L \arrow[r, "\langle C\times L, n\circ\pi'\rangle"] \arrow[d, "\pi'"'] & C\times L\times N \arrow[r, "\mathtt{Ti(shunt}(\alpha))"] \arrow[d, "\pi'\times N"'] & C\times L \arrow[d, "\pi'"] \\
L \arrow[r, "\langle L,n\rangle"'] & L\times N \arrow[r, "\mathtt{Ti(tail)}"'] & L
\end{tikzcd}$$

commutes since π' is a loop morphism from $\mathtt{shunt}(\alpha)$ to $\mathtt{tail}$. Its lower edge is $\mathtt{nil}$ by assumption and so $\mathtt{shunt}(\alpha)^{n\circ\pi'}$ factors through the fixed subobject $\langle C, \mathtt{nil}\rangle : C \rightarrow C\times L$ of $\mathtt{shunt}(\alpha)$ as required. $\square$

8 Acknowledgements

My thanks go to Robin Cockett for several fruitful discussions. I have also profited from stimulating conversations with many people, particularly, Rod Burstall, Pierre-Louis Curien, Peter Freyd, Martin Hofmann, John Hughes, Peter Johnstone, Geraint Jones, Ed Kazmierczak, Francois Lamarche, Phil Mulry, Gordon Plotkin, Leopoldo Roman, Phil Scott, Mary Sheeran, and the members of the Applications of Categories Club. Especial thanks are due to Pierre-Louis for his patience and support in the preparation of this paper.

References

[1] M. Barr, $*$-autonomous categories and linear logic, Mathematical Structures in Computer Science 1(2) (1991) 159–178.

[2] M. Barr and C. Wells, *Toposes, Triples and Theories* (Springer-Verlag, 1984).

[3] R. Bird and P. Wadler, *Introduction to Functional Programming* International Series in Computer Science, ed: C.A.R. Hoare (Prentice Hall, 1988).

[4] A. Carboni, Bicategories of partial maps, Cah. de Top. et Géom. Diff. 28(2) (1987).

[5] J.R.B. Cockett, List-arithmetic distributive categories: locoi, J. Pure and Appl. Alg. 66 (1990) 1–29.

[6] J.R.B. Cockett, Conditional Control is Not Quite Categorical Control, in: G. Birtwistle (ed), *IV Higher Order Workshop, Banff, 1990,* (Springer, 1991) 190–217.

[7] J.R.B. Cockett, Strong Categorical Datatypes I, in: M. Fourman et al (eds) *Proceedings of the Durham Symposium on Applications of Categories in Computer Science,* to appear.

[8] R.L. Crole and A.M. Pitts, New Foundations for Fixpoint Computations, in: J.C. Mitchell et al (eds), *Proceedings Fifth Annual IEEE Symposium on Logic in Computer Science* (IEEE Computer Society Press, 1990) 489–497.

[9] T. Fox, Coalgebras and cartesian categories, Communications in Algebra 4 (7) (1976) 665–667.

[10] C.B. Jay, A note on natural numbers objects in monoidal categories, Studia Logica 48(3) (1989) 389–393.

[11] C.B. Jay, Partial functions, ordered categories, limits and cartesian closure, in: G. Birtwistle (ed), *IV Higher Order Workshop, Banff, 1990,* (Springer, 1991).

[12] C.B. Jay, Fixpoint and loop constructions as colimits, in: A. Carboni et al (eds), *Category Theory: Proceedings, Como 1990* Lecture Notes in Mathematics 1488 (Springer-Verlag, 1991) 187–192.

[13] C.B. Jay, Tail recursion from universal invariants, in: D.H. Pitt et al (eds), *Category Theory and Computer Science Paris, France, September 1991 Proceedings,* Lecture Notes in Computer Science 530 (Springer, 1991) 151–163.

[14] C.B. Jay, Tail recursion via universal invariants, Tech. Rep. ECS–LFCS–91–151.

[15] C.B. Jay, Matrices, monads and the Fast Fourier Transform, in preparation.

[16] G. Jones, Calculating the Fast Fourier Transform, in: G. Birtwistle (ed), IV Higher Order Workshop, Banff, 1990, (Springer-Verlag, 1991).

[17] W. Khalil and R.F.C. Walters, An imperative language based on distributive categories II, Tech. Rep. Pure Mathematics 91-29, The University of Sydney.

[18] G.M. Kelly and R. Street, Review of the elements of 2-categories, Lecture Notes in Mathematics 420 (Springer-Verlag, 1974) 75–103.

[19] G.M. Kelly, Basic concepts of enriched category theory, Lecture Note Series 64, (London Math. Soc. 1982).

[20] G.M. Kelly, On clubs and data-type constructors, in: M. Fourman et al (eds) *Proceedings of the Durham Symposium on Applications of Categories in Computer Science,* to appear.

[21] A. Kock, Monads on symmetric monoidal closed categories, Archiv der Mathematik 21 (1970).

[22] J. Lambek, A fixpoint theorem for complete categories, Math. Zeitschr. 103 (1968), 151–161.

[23] J. Lambek, Fixpoints revisited, Lecture Notes in Computer Science 363 (Springer-Verlag, 1989) 200–207.

[24] J. Lambek and P. Scott, *Introduction to higher order categorical logic,* Cambridge studies in advanced mathematics 7, Cam. Univ. Press (1986).

[25] F.W. Lawvere, Adjointness in Foundations, *Dialectica* 23 (1969) 281–296.

[26] F.W. Lawvere, Metric Spaces, Generalized Logic, and Closed Categories, Rend. del Sem. Mat. e Fis. di Milano 43 (1973) 135–166.

[27] F.W. Lawvere and S.H. Schanuel, *Conceptual Mathematics: A First Introduction to Categories (preliminary version).*

[28] S. Mac Lane, *Categories for the Working Mathematician* (Springer-Verlag, 1971).

[29] J. Meseguer, Completions, factorisations and colimits for ω-posets, in *Mathematical Logic in Computer Science*, Colloquia Mathematica Janos Bolyai, XXVI (Hungary, 1978) 509–545.

[30] E. Moggi, Computational lambda-calculus and monads, Proceedings Fourth Annual Symposium on Logic in Computer Science (IEEE Computer Society Press, 1989) 14–23.

[31] P.S. Mulry, Categorical Fixed Point Semantics, Theoretical Computer Science 70 (1990) 85–97.

[32] R. Paré and L. Roman, Monoidal categories with natural numbers object, Studia Logica 48(3) (1989).

[33] G. Rosolini and E. Robinson, *Categories of partial maps*, Inf. and Comp. 79(2) (1988) 95–130.

[34] L. Roman, Cartesian categories with natural numbers object, J. of Pure and Appl. Alg. 58 (1989) 267–278.

[35] R.A.G. Seely, Linear logic, $*$-autonomous categories and cofree coalgebras, in: J. Gray and A. Scedrov (eds), *Categories in Computer Science and Logic* (Proc. A.M.S. Summer Research Conference, June 1987), Contemporary Mathematics 92, (Am. Math. Soc. 1989).

[36] M.B. Smyth & G.D. Plotkin, The Category-theoretic solution of recursive domain equations, SIAM J. of Comp. 11 (1982).

[37] R.F.C. Walters, Datatypes in distributive categories, Bull. Australian Math. Soc. 40 (1989) 79–82.