Calculation Properties of Initiality

Maarten M Fokkinga
CWI, Amsterdam, and University of Twente, Enschede

Version of April 9, 1991

We give a systematic treatment of the calculation properties brought forward by initiality, and finality.

1 Introduction

Many concepts in Category Theory are defined by so-called universal properties, in particular initiality and, dually, finality. The fact that an object is initial in a category gives immediately several algebraic properties that are quite attractive from a calculational point of view. In most texts on category theory these properties are not made explicit; this is related to the fact that in those texts the proofs are not calculational (algebraic), but pictorial. This is remarkable since category theory might be viewed as an algebraic formalisation of common structures in many branches of mathematics: why should the categorician not work in an algebraic fashion herself?

We feel that algebraic proofs are superior to pictorial proofs for several reasons; for example: precision, verifiability, presentation, possibility for machine assistence, and so on. Pictures may be helpful in presenting the data involved (the morphisms) in an organised way, but the need for this is somewhat reduced by good and suggestive names for the various unique morphisms that exist on account of initiality; such naming is just one aspect of our proposal here. Pictures may also suggest some ways to go, but well designed formulas do so too.

In Section 2 we define initiality and list its algebraic properties. Then in Section 3 we specialise the definition and properties to initial algebras and coproducts, coequalisers and kernel pairs, and finally colimits in general, and we show the algebraic properties in action by proving some well-known facts. We shall present our calculations in fine-grained steps; an experienced calculator will presumably do several steps at once. The large amount of space that our calculations (with hints!) take, is compensated for by the absence of pictures of triangles and rectangles.

Notation Formula $x : a \to_{\mathbf{C}} b$ is the statement that x is a morphism in category $\mathbf{C}$ with source a and target b. We denote composition in diagramatic order: if $x : a \to b$

and $y : b \to c$ then $x ; y : a \to c$. We use $a, b, c, \ldots$ for objects in $\mathbf{C}$, $F, G, H \ldots, I, K, \mathit{0}, S$ for functors, and $\gamma, \delta, \epsilon, \eta$ for natural transformations.

Remark It is the morphisms in a category that are important; the objects only serve an auxiliary role, mainly for the well-formedness of morphism composition. (Formally, objects are superfluous since their role can be taken over by the identity morphisms.) In any category $\mathbf{C}$, including those of functors on $\mathbf{C}$ and so on, any morphism has a unique source and target, so that you need not mention them explicitly; they can be retrieved from the morphism itself by the functions $\mathrm{src}_{\mathbf{C}}(\_)$ and $\mathrm{tgt}_{\mathbf{C}}(\_)$. Thus we view a functor mainly as a mapping of morphisms; its action on objects can then be derived. In the same vein, we shall define concepts in terms of morphisms as much as possible, and suppress the auxiliary role of objects when it can be derived or is clear from the context. □

2 Initiality

Let $\mathbf{C}$ be a category, an a an object in $\mathbf{C}$. Then a is *initial in* $\mathbf{C}$ if:

$$x : a \to_{\mathbf{C}} b \qquad \equiv \qquad x = (\!|b|\!)_{\mathbf{C},a} \qquad \text{CHARN}$$

Here $(\!|\ |\!)$ is a notation for a morphism (depending only on $\mathbf{C}, a$ and b), and all free variables in the line are understood to be universally quantified, except those that have been introduced in the immediate context ($\mathbf{C}$ and a in this case). Charn is mnemonic for Characterisation. Thus reading the equivalence from left to right, it says that each morphism x with a as its source, is uniquely determined by its target b (if it exists at all). Moreover, reading from right to left and taking $x := (\!|b|\!)_{\mathbf{C},a}$ we find that for each b there is a morphism from a to b. In view of the many algebraic properties that the unique morphism from a to b has, it is a good thing to have a name for it (so that we can express that property without being forced first to invent a name). The brackets "$(\!|\ |\!)$" do just that, and they differ enough from overbars, primes, dashes and the like so as to clearly signal the special properties listed below. In some cases there is a more specific notation that better suggests the resulting properties (see coequalisers and kernel pairs below).

Of course, when $\mathbf{C}$ is clear from the context we omit subscript $\mathbf{C}$. It often happens that one initial object in $\mathbf{C}$ is fixed, and in that case subscript a can also be dropped. For the dual case, the default notation for the unique morphism from b to the final object a is $[\![b]\!]_{\mathbf{C},a}$.

Let us now give some consequences of Charn. By a substitution for x such that the right-hand side becomes true we find Self, and by a substitution for b, x such that the left-hand side becomes true we find Id:

$$(\!|b|\!)_a : a \to b \qquad \text{SELF}$$

$$id = (\!|a|\!)_a \qquad \text{ID}$$

Next we have the Uniqueness and Fusion property:

$$x, y : a \to b \quad \Rightarrow \quad x = y \qquad \textsc{Uniq}$$

$$x : b \to c \quad \Rightarrow \quad (\!|b|\!)_a ; x = (\!|c|\!)_a \qquad \textsc{Fusion}$$

The proof of Uniq is left to the reader. For Fusion we argue:

$$\begin{array}{ll} & (\!|b|\!)_a ; x = (\!|c|\!)_a \\ \equiv & \qquad \textsc{Charn}[b, x := c, (\!|b|\!)_a ; x] \\ & (\!|b|\!)_a ; x : a \to c \\ \Leftarrow & \qquad \text{composition} \\ & (\!|b|\!)_a : a \to b \quad \wedge \quad x : b \to c \\ \equiv & \qquad \textsc{Self}, \text{ and premise} \\ & \mathsf{true}. \end{array}$$

The importance of law Fusion cannot be over-emphasised; we shall use it below quite often. You may also read Fusion as giving a sufficient condition on x and b in order that the composite $(\!|b|\!)_a ; x$ can be expressed as a morphism of the form $(\!|...|\!)_a$. In the case of initial algebras Uniq captures the pattern of proofs by induction that two functions x and y are equal; see below.

In the sequel we shall meet categories $\mathcal{C}(\mathbf{C})$ built upon $\mathbf{C}$ in such a way that an object in $\mathcal{C}(\mathbf{C})$ is a morphism in $\mathbf{C}$, and a morphism $x : a \to_{\mathcal{C}(\mathbf{C})} b$ in $\mathcal{C}(\mathbf{C})$ is just a morphism x in $\mathbf{C}$ that, in compositions with a and b, satisfies a certain equation. The above laws become then less trivial and much more interesting, although the structure remains the same.

In some cases, depending on the way formula $x : b \to_{\mathcal{C}(\mathbf{C})} c$ is expressed in compositions of $\mathbf{C}$, we can derive immediately some further useful algebraic properties. For example, for coproducts and coequalisers it turns out that formula $x : b \to_{\mathcal{C}(\mathbf{C})} c$ expresses c in b and x. In such a case Fusion can be formulated as an unconditional law (by substituting the expression for c in the conclusion). In this case it may also happen that Uniq boils down to the assertion that a is epi. And, again depending on the form of $x : a \to_{\mathcal{C}(\mathbf{C})} b$, we can sometimes also derive that an initial object a is monic as well (by a proper use of Self), see the discussion of coproducts below.

Experience shows that many a proof that is done pictorially, can now be given calculationally. And, more importantly, many a proof can be constructed by reducing the proof obligation step by step to true, using algebraic manipulations, and using the form of the formulas as the driving force, like we did above for Fusion.

Application Here is a first example of the use of these laws: the proof of the fact that an initial object is unique up to a unique isomorphism. Suppose that both a and b are initial. We claim that $(\!|b|\!)_a$ and $(\!|a|\!)_b$ establish the isomorphism and are unique in doing so. First we show the correct typing.

$$
\begin{array}{ll}
& (\!|b|\!)_a : a \to b \qquad \wedge \qquad (\!|a|\!)_b : b \to a \\
\equiv & \qquad \text{SELF on both conjuncts} \\
& \mathsf{true}.
\end{array}
$$

Next we show

$$x = (\!|b|\!)_a \ \wedge \ y = (\!|a|\!)_b \qquad \equiv \qquad x ; y = id_a \ \wedge \ y ; x = id_b$$

that is, their composition is the identity, and conversely the identities can be factored only in this way. So we prove both implications of the equivalence at once.

$$
\begin{array}{ll}
& x = (\!|b|\!)_a \qquad \wedge \qquad y = (\!|a|\!)_b \\
\equiv & \qquad \text{CHARN} \\
& x : a \to b \qquad \wedge \qquad y : b \to a \\
\equiv & \qquad \text{composition} \\
& x ; y : a \to a \qquad \wedge \qquad y ; x : b \to b \\
\equiv & \qquad \text{CHARN} \\
& x ; y = (\!|a|\!)_a \qquad \wedge \qquad y ; x = (\!|b|\!)_b \\
\equiv & \qquad \text{ID} \\
& x ; y = id_a \qquad \wedge \qquad y ; x = id_b.
\end{array}
$$

The industrious reader may show $(\!|b|\!)_a ; (\!|a|\!)_b = id_a$ alternatively using ID, FUSION, and SELF in that order. (Together with the symmetrical $(\!|a|\!)_b ; (\!|b|\!)_a = id_b$ and the uniqueness of the factorisation of id_a and id_b, you have an alternative proof.)

3 Specialisations

Initial algebras

Let $\mathbf{C}$ be a category, and F be a endofunctor on $\mathbf{C}$. Unless stated otherwise explicitly, all morphisms and objects are in $\mathbf{C}$. A morphism φ is called an *F-algebra (over $\mathbf{C}$)* if $\varphi : Fa \to a$ for some object a, called its *carrier*. A morphism h is called an *F-algebra homomorphism* if for some F-algebras φ and ψ, $\varphi ; h = Fh ; \psi$ (and therefore $h : a \to b$ where a is the carrier of φ and b the carrier of ψ). The category $\mathrm{Alg}_{\mathbf{C}}(F)$ has the F-algebras over $\mathbf{C}$ as its objects, the F-algebra homomorphisms as its morphisms (with source and target defined in the obvious way), and inherits the composition from $\mathbf{C}$ in the obvious way. All this should be well known; an extensive motivation for this definition is given by Fokkinga and Meijer [1]. We write $x : \varphi \to_F \psi$ instead of $x : \varphi \to_{\mathrm{Alg}(F)} \psi$. Now let α be initial in $\mathrm{Alg}(F)$; we fix this α throughout what follows, and suppress its occurrences as a subscript. (Note that α is the *constructor of the data type*.) Then the laws of initiality work out as follows.

$$
\begin{array}{lcll}
x : \alpha \rightarrow_F \varphi & \equiv & x = (\!|\varphi|\!)_F & \text{CHARN} \\
(\!|\varphi|\!)_F : \alpha \rightarrow_F \varphi & & & \text{SELF} \\
id_\alpha = (\!|\alpha|\!)_F & & & \text{ID} \\
x, y : \alpha \rightarrow_F \varphi & \Rightarrow & x = y & \text{UNIQ} \\
x : \varphi \rightarrow_F \psi & \Rightarrow & (\!|\varphi|\!)_F ; x = (\!|\psi|\!)_F & \text{FUSION}
\end{array}
$$

Remember that in all these laws formula $x : \alpha \rightarrow_F \varphi$ means that x satisfies the recursion equation $\alpha ; x = Fx ; \varphi$, or $x = \alpha^\cup ; Fx ; \varphi$ where $\alpha^\cup$ is the inverse of α that we shall construct below. Thus CHARN says that the "inductive definition" $x = \alpha^\cup ; Fx ; \varphi$ has a unique solution. Similarly, UNIQ says that if two functions x and y both satisfy the same "inductive pattern", namely $x = \alpha^\cup ; Fx ; \varphi$ and $y = \alpha^\cup ; Fy ; \varphi$, then they are the same. One sees that UNIQ captures, in a sense, induction.

Application As an example calculation let us show the well-known fact that the initial F-algebra α is —up to isomorphism— a fixed point of F, i.e., $\alpha \simeq F\alpha$ in $\mathrm{Alg}(F)$. This requires us to establish a pair x, y of morphisms in $\mathrm{Alg}(F)$ (F-algebra homomorphisms in $\mathbf{C}$),

$$
\begin{array}{lcl}
x & : & \alpha \rightarrow_F F\alpha \\
y & : & F\alpha \rightarrow_F \alpha\,,
\end{array}
$$

that are each others inverse. Since initiality of α in $\mathrm{Alg}(F)$ is given, we know by the proof of the uniqueness of initial objects that the only choice for x is $(\!|F\alpha|\!)_F$. By unfolding the requirement for y we immediately find that the only choice for y is α. It remains to show that these choices are each others inverse indeed. For this we argue:

$$
\begin{array}{ll}
& (\!|F\alpha|\!)_F ; \alpha = id \\
\equiv & \quad \text{ID} \\
& (\!|F\alpha|\!)_F ; \alpha = (\!|\alpha|\!)_F \\
\Leftarrow & \quad \text{FUSION} \\
& \alpha : F\alpha \rightarrow_F \alpha \\
\equiv & \quad \text{argued above, or: definition } \rightarrow_F \\
& \mathsf{true}.
\end{array}
$$

So $(\!|F\alpha|\!)_F$ is a pre-inverse of α. It is a post-inverse too:

$$
\begin{array}{ll}
& \alpha ; (\!|F\alpha|\!)_F \\
= & \quad \text{SELF}
\end{array}
$$

$$
\begin{array}{ll}
& F(\!|F\alpha|\!)_F ; F\alpha \\
= & \quad \text{functor, above: } (\!|F\alpha|\!)_F \text{ is pre-inverse of } \alpha \\
& Fid.
\end{array}
$$

(One should compare this calculation with the usual pictorial proofs, and in particular pay attention to the precision with which various steps in the proof are stated.) Putting $a := \mathrm{src}(\alpha)$ (so that $Fa = \mathrm{src}(F\alpha)$) we have now as a corollary also the isomorphism $a \simeq Fa$ in category $\mathbf{C}$.

Many more examples of the use of these laws have been given by Fokkinga and Meijer [1].

Coproducts

Coproducts can be described in various ways, for example as initial K_a, K_b-algebras (where K_a is the constant endofunctor mapping each morphism on id_a, and generalising the notion of F-algebra to F, G-algebra), or as colimits of the diagram $\{id_a, id_b\}$ (see below). In whatever way it is done, the following algebraic properties result. The interested reader may try and derive these laws from her own categorical description of coproduct.

Let inl, inr be a coproduct over a, b. We write $f \triangledown g$ instead of $(\!|f, g|\!)_{(a,b),(inl,inr)}$, thus suppressing the dependency on inl, inr and a, b. (The usual categorical notation is $[f, g]$ and their common source is often written as $a + b$.)

$inl ; x = f \ \wedge \ inr ; x = g$	$\equiv$	$x = f \triangledown g$	Charn
$inl ; f \triangledown g = f \ \wedge \ inr ; f \triangledown g = g$			Self
$inl \triangledown inr = id$			Id
$inl ; x = inl ; y \ \wedge \ inr ; x = inr ; y$	$\Rightarrow$	$x = y$ (inl, inr jointly epi)	Uniq
$f \triangledown g ; x = (f ; x) \triangledown (g ; x)$			Fusion

Application As announced in the general discussion on initiality, we can now easily show that inl is monic (and by symmetry inr too):

$$
\begin{array}{ll}
& x = y \\
\equiv & \quad \text{aiming at "} ; inl \text{" after } x \text{ and } y\text{, use } \textsc{Self}[\, f := id \,] \\
& x ; inl ; id \triangledown g = y ; inl ; id \triangledown g \\
\Leftarrow & \quad \text{Leibniz} \\
& x ; inl = y ; inl
\end{array}
$$

as desired. The choice for g is immaterial; id certainly does the job.

These laws are ubiquitous in the work by Fokkinga and Meijer [1].

Coequalisers

Coequalisers form an interesting example since their additional properties suggest a more specific notation for the unique morphisms established by initiality. We are dealing here with a special case of colimits discussed below.

Let us start afresh in an arbitrary category $\mathbf{C}$. Unless explicitly said otherwise, all morphisms and objects are in $\mathbf{C}$, and we omit everywhere subscript $\mathbf{C}$. A *fork* is a triple f, g, p of morphisms such that $f;p = g;p$, and hence $\mathrm{tgt}(f) = \mathrm{tgt}(g) = a = \mathrm{src}(p)$ for some a; we call (f, g) an a*-prong* and p an a*-haft*. (So, an a-prong is just a parallel pair with target a, and an a-haft is just a morphism with source a.)

Picture 1. An a-fork (f, g, p) with a-prong (f, g) and a-haft p.

A morphism h is an a-haft homomorphism if $p;h = q$ for some a-hafts p and q (and hence $h : \mathrm{tgt}(p) \to \mathrm{tgt}(q)$). This determines a category $\mathrm{Haft}(a)$ in the obvious way; and we write $h : p \to_a q$ instead of $h : p \to_{\mathrm{Haft}(a)} q$. For a-prong (f, g) we define $\mathrm{Haft}(f, g)$ to be the full subcategory of $\mathrm{Haft}(a)$ whose objects form a fork with f, g, i.e., the objects are p with $f;p = g;p$. An object p that is initial in $\mathrm{Haft}(f, g)$ is called a *coequaliser of* (f, g).

We list again the algebraic properties brought forward by initiality of coequalisers. This time we write $p\backslash_{f,g}q$ or simply $p\backslash q$ instead of $(\!|q|\!)_{(f,g),p}$, since, as we shall explain, the fraction notation is a better suggestion for its calculational properties. In the laws we assume that p is a coequaliser of (f, g). We spell out formulas $x : q \to_{f,g} r$ as $q;x = r$ (which implies that r is an (f, g)-haft if q is so).

	$p;x = q$	$\equiv$	$x = p\backslash q$	CHARN
	$p;p\backslash q = q$			SELF
	$id = p\backslash p$			ID
	$p;x = q \ \wedge\ p;y = q$	$\Rightarrow$	$x = y$	UNIQ
i.e.,	$p;x = p;y$	$\Rightarrow$	$x = y$	(p epi)
	$q;x = r$	$\Rightarrow$	$p\backslash q;x = p\backslash r$	FUSION
i.e.,	$p\backslash q;x = p\backslash(q;x)$			

In the laws CHARN, SELF and FUSION we have omitted the premise that q is an (f, g)-haft; this is rather harmless if we agree that the notation $...\backslash q$ requires a separate proof that q is an (f, g)-haft. The nice, unconditional, form of the fusion law is due to the form of $x : q \to_{f,g} r$ when spelled out in terms of composition within category $\mathbf{C}$: in $q; x = r$ we can "solve" r and substitute its solution in the conclusion of the law. Due to a similar reason law UNIQ gets the form of the assertion that a coequaliser is epi.

Now that we have presented the laws the choice of notation may be evident: the usual manipulation of *cancelling adjacent factors in the denominator and nominator* is valid when composition is interpreted as multiplication and $\backslash$ is interpreted as a fraction. This may also help you to remember you that there is only post-fusion here; the equation $x; p\backslash q = (x; p)\backslash q$ is not valid. (In the dual case, of course, there is only pre-fusion, see the kernel pairs below.)

Additional laws As the reader may expect, the following law is valid too:

$$p\backslash q; q\backslash r = p\backslash r \qquad \text{COMPOSE}$$

One can prove this in various ways; here is one way:

$$\begin{array}{ll} & p\backslash q; q\backslash r \\ = & \qquad \text{FUSION} \\ & p\backslash (q; q\backslash r) \\ = & \qquad \text{SELF} \\ & p\backslash r. \end{array}$$

The interesting aspect here is that the suppressed subscripts to $\backslash$ may differ: e.g., $p\backslash_{f,g} q$ and $q\backslash_{h,j} r$, and q is not necessarily a coequaliser of f, g. Rephrased in terms of Section 2 we have:

$$(\!|b|\!)_{\mathcal{A},a}; (\!|c|\!)_{\mathcal{B},b} = (\!|c|\!)_{\mathcal{A},a} \qquad \text{COMPOSE}$$

where $\mathcal{A}$ and $\mathcal{B}$ are full subcategories of some category $\mathcal{C}(\mathbf{C})$ and objects b, c are in both $\mathcal{A}$ and $\mathcal{B}$; in our case $\mathcal{A} = \text{Haft}(f,g)$ say, $\mathcal{B} = \text{Haft}(h,j)$, and $\mathcal{C}(\mathbf{C}) = \text{Haft}(a)$. The proof runs then as follows.

$$\begin{array}{ll} & (\!|b|\!)_{\mathcal{A},a}; (\!|c|\!)_{\mathcal{B},b} = (\!|c|\!)_{\mathcal{A},a} \\ \Leftarrow & \qquad \text{FUSION} \\ & (\!|c|\!)_{\mathcal{B},b} : b \to_{\mathcal{A}} c \\ \equiv & \qquad \text{both } \mathcal{A} \text{ and } \mathcal{B} \text{ full subcategories of } \mathcal{C}(\mathbf{C}), \\ & \qquad \text{each containing both } b \text{ and } c \\ & (\!|c|\!)_{\mathcal{B},b} : b \to_{\mathcal{B}} c \\ \equiv & \qquad \text{SELF} \\ & \mathsf{true}. \end{array}$$

Another law that we shall use below has to do with functors.

$$\text{if } F \text{ preserves coequalisers, then} \qquad F(p\backslash q) = Fp\backslash Fq \qquad \text{FCTR1}$$

if $Fp\backslash ...$ is well-defined, then $\qquad F(p\backslash q) = Fp\backslash Fq \qquad$ FCTR2

Clearly, FCTR1 is a special case of FCTR2. To prove the latter we argue

$$
\begin{array}{ll}
 & F(p\backslash q) = Fp\backslash Fq \\
\equiv & \qquad \text{CHARN} \\
 & Fp ; F(p\backslash q) = Fq \\
\equiv & \qquad \text{functor} \\
 & F(p ; p\backslash q) = Fq \\
\equiv & \qquad \text{ID} \\
 & \mathsf{true}.
\end{array}
$$

Application Here is an illustration of these laws. Suppose that p is a coequaliser of an a-prong (f,g), and $p = q ; r$ where r is monic. Then we can construct a pre-inverse of r, namely $p\backslash q$. First we show that $p\backslash q$ exists, i.e., q is an (f,g)-haft.

$$
\begin{array}{ll}
 & f ; q = g ; q \\
\Leftarrow & \qquad r \text{ monic} \\
 & f ; q ; r = g ; q ; r \\
\equiv & \qquad \text{introduction } q,r \text{; and } p \text{ is an } (f,g)\text{-haft} \\
 & \mathsf{true}.
\end{array}
$$

Next we show that $p\backslash q$ is a pre-inverse of r.

$$
\begin{array}{ll}
 & p\backslash q ; r \\
= & \qquad \text{FUSION} \\
 & p\backslash (q ; r) \\
= & \qquad \text{introduction } q,r \\
 & p\backslash p \\
= & \qquad \text{ID} \\
 & id.
\end{array}
$$

In **Set** the coequaliser p of (f,g) represents the least equivalence relation $R_{f,g}$ that contains $\{z :: (f\,z, g\,z)\}$, in the sense that $(=_p) = R_{f,g}$ (where we define $x =_p y \equiv p(x) = p(y)$). Any equivalence relation $\simeq$ can be written as such a $R_{f,g}$. For example, take the set $\{x,y : x \simeq y : (x,y)\}$ as the common source of f and g, and take $f,g = \pi_0, \pi_1$; then $R_{f,g} = (\simeq)$. In general there are many more possibilities; the particular choice for f,g just given is known as the kernel pair of p.

Kernel pairs

As usual everything is relative to a category $\mathbf{C}$, without mentioning this explicitly. We use the terminology of *fork*, *haft*, and *prong* introduced in the discussion of coequalisers. For a-prongs (f,g) and (h,j) we define x to be an a-prong-homomorphism from (f,g) to (h,j) if $f = x;h$ and $g = x;j$. Thus we have a category $\mathrm{Prong}(a)$ that has a-prong homomorphisms as its morphisms (and inherits its composition from $\mathbf{C}$). We abbreviate as usual $\to_{\mathrm{Prong}(a)}$ to $\to_a$. $\mathrm{Prong}(p)$ is the full subcategory of prongs (f,g) that together with p form a fork, i.e., $f;p = g;p$. A final object in $\mathrm{Prong}(p)$ is called a *kernel pair of* p. This time we use the notation $(k,l)/_p(f,g)$ instead of $[\![k,l]\!]_{p,(f,g)}$.

The algebraic properties of a p-kernel pair f,g are as follows. (We have already worked out the formulas in terms of $\mathbf{C}$, and performed the simplification in Uniq and Fusion.)

$$
\begin{array}{llcll}
 & k = x;f \ \wedge\ l = x;g & \equiv & x = (k,l)/(f,g) & \textsc{Charn} \\
 & k = (k,l)/(f,g);f \quad \wedge \quad l = (k,l)/(f,g);g & & & \textsc{Self} \\
 & id = (f,g)/(f,g) & & & \textsc{Id} \\
 & \begin{array}{l} k = x;f \ \ \wedge\ \ l = x;g \\ k = y;f \ \ \wedge\ \ l = y;g \end{array} & \Rightarrow & x = y & \textsc{Uniq} \\
\text{i.e.,} & x =_f y \quad \wedge \quad x =_g y & \Rightarrow & x = y & ((f,g) \text{ jointly monic}) \\
 & x;(k,l)/(f,g) = (x;k,\ x;l)/(f,g) & & & \textsc{Fusion} \\
 & (f,g)/(h,j);(h,j)/(k,l) = (f,g)/(k,l) & & & \textsc{Compose} \\
 & \text{if } F \text{ preserves kernel pairs, then } F((f,g)/(h,j)) = F(f,g)/F(h,j) & & & \textsc{Fctr1} \\
 & \text{if } F(f,g)/... \text{ is well-defined, then } F((f,g)/(h,j)) = F(f,g)/F(h,j) & & & \textsc{Fctr2}
\end{array}
$$

Notice that there is pre-fusion only.

Application As an example of the use of these laws we prove that the coequaliser and kernel pair form an adjunction. To be precise, let C denote a mapping that sends any a-prong to some coequaliser of the prong, and similarly let K send any a-haft to some kernel pair of the haft. We extend these mapping to functors $C : \mathrm{Prong}(a) \to \mathrm{Haft}(a)$ and $K : \mathrm{Haft}(a) \to \mathrm{Prong}(a)$, by defining

$$
\begin{array}{lclcll}
Cu & = & C(d,e)\backslash C(f,g) & : & C(d,e) \to_a C(f,g) & \text{for } u : (d,e) \to_a (f,g) \\
Kx & = & Kp/Kq & : & Kp \to_a Kq & \text{for } x : p \to_a q
\end{array}
$$

We shall establish natural transformations $\epsilon : CK \stackrel{.}{\to} I$ and $\eta : I \stackrel{.}{\to} KC$ such that $\eta K;K\epsilon = idK$ and $C\eta;\epsilon C = Cid$. Take $\epsilon_q = CKq\backslash q : CKq \to_a q$ for all q in $\mathrm{Haft}(a)$. Then the naturality is shown as follows. For arbitrary $u : q \to_a r$

$$
\begin{array}{ll}
 & CKu;\epsilon_r \\
= & \quad \text{definition } C,\ K \text{ and } \epsilon, \text{ noting that } u : q \to_a r
\end{array}
$$

$$
\begin{array}{ll}
 & CKq\backslash CKr \,;\, CKr\backslash r \\
= & \qquad \textsc{Compose} \\
 & CKq\backslash r \\
= & \qquad \text{definition haft homomorphism, } u : q \to_a r \\
 & CKq\backslash (q\,;\, u) \\
= & \qquad \textsc{Fusion} \\
 & CKq\backslash q \,;\, u \\
= & \qquad \text{definition } \epsilon \text{ and } I \\
 & \epsilon_q \,;\, Iu
\end{array}
$$

as desired. Further we take $\eta_{(h,j)} = (h,j)/KC(h,j) : (h,j) \to_a KC(h,j)$. We omit the proof that η is natural; this is quite similar (but not categorically dual) to the naturality of ϵ. It remains to show that $\eta K \,;\, K\epsilon = idK$. Let q be arbitrary, then

$$
\begin{array}{ll}
 & (\eta K \,;\, K\epsilon)_q \\
= & \qquad \text{definitions} \\
 & Kq/KCKq \,;\, K(CKq\backslash q) \\
= & \qquad \textsc{Fctr2} \\
 & K(q/CKq) \,;\, K(CKq/q) \\
= & \qquad \text{functor, } \textsc{Compose} \\
 & K(q/q) \\
= & \qquad \textsc{Id} \\
 & idK.
\end{array}
$$

And the proof of $C\eta \,;\, \epsilon C = Cid$ is again quite similar to the above one.

Colimits

An initial object is just a colimit of the empty diagram, and conversely, a colimit of a diagram is just an initial object in the category of cocones over that diagram. Let us therefore present the algebraic properties of colimits.

A *diagram* D is a collection of morphism. The sources and targets of members of D are collectively called its objects. A collection $\gamma = \{a \text{ object in } D :: \gamma_a\}$ of morphisms with common target (called the target of γ) is called a D-*cocone* if

$$\forall (f : a \to b \text{ in } D :: \quad \gamma_a = f \,;\, \gamma_b).$$

(A cocone must have a distinguished target even if it is empty.) A morphism h is called a D-cocone homomorphism, from D-cocone γ to D-cocone δ say and denoted $h : \gamma \to_D \delta$, if

$$\forall (a \text{ in } D :: \quad \gamma_a \,;\, h = \delta_a).$$

This determines a category $\mathrm{Cocone}(D)$ in the usual way, and an initial object in it is called a *colimit for* D *in* $\mathbf{C}$. Writing $\gamma\backslash_D\delta$ or simply $\gamma\backslash\delta$, instead of $(\!|\delta|\!)_{D,\gamma}$, we have the following algebraic properties. In the formulation we assume that D is a diagram, and that γ is a colimit for D. (We have spelled out the statements involving $\to_D$; a ranges over the objects of D.)

$\forall(a :: \quad \gamma_a ; x = \delta_a)$	$\equiv$	$x = \gamma\backslash\delta$	CHARN
$\forall(a :: \quad \gamma_a ; \gamma\backslash\delta = \delta_a)$			SELF
$id = \gamma\backslash\gamma$			ID
$\forall(a :: \quad \gamma_a ; x = \gamma_a ; y)$	$\Rightarrow$	$x = y \quad (\gamma \text{ epi})$	UNIQ
$\gamma\backslash\delta ; x = \gamma\backslash\{a :: \ \delta_a ; x\}$			FUSION
$\gamma\backslash\delta ; \delta\backslash\epsilon = \gamma\backslash\epsilon$			COMPOSE
if $F\gamma\backslash\ldots$ defined, then		$F(\gamma\backslash\delta) = F\gamma\backslash F\delta$	FCTR2

for any D-cocones δ, ϵ (δ being a colimit in law COMPOSE).

Improved description In view of the explicit quantifications the above laws for colimits are not very suited for algebraic, equational calculation. We can avoid a lot of explicit quantifications by treating a cocone as a family of functions, and defining for example $\gamma ; x = \delta$ to mean $\forall(a :: \ \gamma_a ; x = \delta_a)$. It turns out that we can perform this trick in a categorical fashion by using natural transformations, which are families of morphisms indeed. Several (not all) manipulations on the subscripts can then be phrased as well-known manipulations with natural transformations as a whole. So let us redesign the definitions. (I've learned this from Jaap van der Woude recently.)

As regards to the property of being a cocone we can assume without loss of generality that a diagram is a subcategory: just add all the compositions of composable arrows, and all the appropriate identities. Going one step further we can consider the subcategory to be the image under a functor $D : \mathbf{D} \to \mathbf{C}$, where $\mathbf{D}$ is a category that gives the shape of the diagram. So we define: a *diagram* in $\mathbf{C}$ is a functor $D : \mathbf{D} \to \mathbf{C}$, for some category $\mathbf{D}$, called the *shape* of the diagram. Further: a *cocone for* D is a natural transformation $\gamma : D \dot{\to} K_c$ for some object c in $\mathbf{C}$ (K_c is the constant functor, $K_c x = id_c$). Indeed, for any $Df : Da \to Db$ in the 'diagram' $D\mathbf{D}$ in $\mathbf{C}$ we find

$$\begin{array}{lllll} & \gamma_a ; K_c f & = & Df ; \gamma_b & : \quad Da \to K_c \\ \text{i.e.,} & \gamma_a & = & Df ; \gamma_b & : \quad Da \to c \end{array}$$

which expresses the required "commutativity of the triangle." (Henceforth we write γa instead of γ_a.) A D-cocone homomorphism is a morphism h such that $\gamma ; h = \delta$, denoted $h : \gamma \to_D \delta$, for some D-cocones γ, δ. This determines a category, called $\mathrm{Cocone}(D)$, and γ is called a *colimit for* D if it is initial in $\mathrm{Cocone}(D)$.

Since cocones are natural transformations, we have the following standard transformations available. For $\gamma : D \dot{\to} K_c$ and $\delta : D \dot{\to} K_d$:

- for any $x : c \to d$,
  $\gamma ; x = \lambda(a :: \ \gamma a ; x) : D \dot{\to} K_d$ is a cocone for D again.

- for any functor $F : \mathbf{C} \to \mathbf{C}$,
  $F\gamma = \lambda(a :: \ F\gamma a) : FD \dot{\to} FK_c$ is a cocone for FD (note that $FK_c = K_{Fc}$).
  If in addition F preserves colimits, then $F\gamma$ is a colimit for FD if γ is so for D.

- for any functor $S : \mathbf{D} \to \mathbf{D}$,
  $\gamma S = \lambda(a :: \ \gamma(Sa)) : DS \dot{\to} K_c$ is a cocone for DS (note that $K_c S = K_c$). If S transforms the shape somewhat, γS is the transformed cocone. Notice that $\gamma S ; \gamma\backslash\delta = (\gamma ; \gamma\backslash\delta)S = \delta S$, but in general $\gamma\backslash\delta \neq \gamma S\backslash\delta S$ since γS need not be a colimit and therefore the latter right-hand side is not well-defined.

Thus for a D-colimit γ we have

$\gamma ; x = \delta$	$\equiv$	$x = \gamma\backslash\delta$	Charn
$\gamma ; \gamma\backslash\delta = \delta$			Self
$\gamma\backslash\gamma = id$			Id
$\gamma ; x = \gamma ; y$	$\Rightarrow$	$x = y$ (γ epi)	Uniq
$\gamma\backslash\delta ; x = \gamma\backslash(\delta ; x)$			Fusion
$\gamma\backslash\delta ; \delta\backslash\epsilon = \gamma\backslash\epsilon$			Compose
if $F\gamma\backslash...$ defined, then		$F(\gamma\backslash\delta) = F\gamma\backslash F\delta$	Fctr2

for any D-cocones δ, ϵ (δ being a colimit in law Compose).

Application We present the well-known construction of an initial F-algebra. Our interest is solely in the algebraic style of various subproofs. We ask the reader to observe that we can be rather concise and clear at the same time (without any need to refer to a picture, or to be verbose by saying "let ... be the unique morphism such that ...") thanks to the formal characterisation Charn of colimits and the standard naming '$\gamma\backslash\delta$' for 'the mediating morphism from colimit γ to cocone δ'.

Given endofunctor F we wish to construct an F-algebra, $\alpha : Fa \to a$ say, that is initial in $\mathrm{Alg}(F)$. Forgoing initiality for the time being, we derive a construction of an $\alpha : Fa \to a$ as follows.

$$
\begin{array}{lll}
 & & \alpha : Fa \to a \\
(a) & \Leftarrow & \quad \text{definition isomorphism} \\
 & & \alpha : Fa \simeq a \\
(b) & \Leftarrow & \quad \text{definition cocone homomorphism (taking } a = \mathrm{tgt}(\gamma) = \mathrm{tgt}(\gamma S)\text{)} \\
 & & \alpha : F\gamma \simeq \gamma S \text{ in } \mathrm{Cocone}(FD) \qquad \wedge \qquad FD = DS \\
(c) & \equiv & \quad F\gamma \text{ is colimit for } FD \text{ (taking } \alpha = F\gamma\backslash\gamma S\text{)}
\end{array}
$$

$$\gamma S \text{ is colimit for } DS \qquad \wedge \qquad FD = DS.$$

Step (a) is motivated by the wish that α be initial in $\mathrm{Alg}(F)$, and so α will be an isomorphism (see the subsection on Initial algebras); in other words, in view of the required initiality the step is *no* weakening. In step (b) we merely decide that α, a come from a (co)limit construction; this is true for virtually all categorical constructions. In order that the step is valid we have to define a diagram D, a D-cocone γ (a colimit say, which we *assume* to exist), and we have to define an endofunctor S on $\mathrm{src}(D)$ (that transforms D-cocone γ into a $DS\ (= FD)$-cocone γS). In step (c) the claim '$F\gamma$ is colimit for FD' follows from the *assumption* that F preserves colimits. The definition $\alpha = F\gamma \backslash \gamma S$ is *forced* by (the proof of) the uniqueness of initial objects. (It is indeed very easy to verify that $F\gamma\backslash\gamma S$ and $\gamma S \backslash F\gamma$ are each others inverse.) We shall now complete the construction in the following three parts.

1. Construction of D, S such that $FD = DS$.
2. Proof of 'γS is colimit for DS' where γ is a colimit for D.
3. Proof of 'α is initial in $\mathrm{Alg}(F)$' where $\alpha = F\gamma\backslash\gamma S$.

Part 1 (Construction of D, S such that $FD = DS$.) The requirement $FD = DS$ says that FD is a 'subdiagram' of D. This is easily achieved by making D a *chain* of iterated F applications, as follows.

Let ω be the category with objects $0, 1, 2, \ldots$ and a unique arrow from i to j (denoted $i{\le}j$) for every $i \le j$. So ω is the shape of a chain. The zero and successor functors $\mathit{0}, S : \omega \to \omega$ are defined by $\mathit{0}(i{\le}j) = 0{\le}0$ and $S(i{\le}j) = (i{+}1){\le}(j{+}1)$.

Let $\mathbf{0}$ be an initial object in $\mathbf{C}$. Define the diagram $D : \omega \to \mathbf{C}$ by $D(i{\le}j) = F^i(\!|F^{j-i}\mathbf{0}|\!)$. It is quite easy to show that D is a functor, i.e., $D(i{\le}j\,;\,j{\le}k) = D(i{\le}j)\,;\,D(j{\le}k)$. It is also immediate that $FD = DS$ since

$$FD(i{\le}j) \;=\; FF^i(\!|F^{j-i}\mathbf{0}|\!) \;=\; F^{i+1}(\!|F^{(j+1)-(i+1)}\mathbf{0}|\!) \;=\; D((i{+}1){\le}(j{+}1)) \;=\; DS(i{\le}j)\,.$$

Part 2 (Proof of 'γS is colimit for DS' where γ is a colimit for D.) Our task is to construct a morphism $(\!|\delta|\!)_{DS,\gamma S}$ such that

$$\gamma S\,;\,x = \delta \quad\equiv\quad x = (\!|\delta|\!)_{DS,\gamma S}$$

for any $\delta : DS \dot{\to} K_d$ (an arbitrary cocone). Our guess is that $\gamma\backslash(\delta', \delta)$ may be chosen for $(\!|\delta|\!)_{DS,\gamma S}$, where $(\delta', \delta) : D \dot{\to} K_d$ is defined by

$$\begin{array}{lclcl cl}
(\delta', \delta)\mathit{0} & = & \delta' & : & D\mathit{0} \dot{\to} K_d\mathit{0} & = & \mathbf{0} \to d \\
(\delta', \delta)S & = & \delta & : & DS \dot{\to} K_dS & = & DS \dot{\to} K_d
\end{array}$$

for some suitably chosen δ'. We prove the desired equivalence, deriving along the way a requirement for δ'.

$$
\begin{array}{lll}
 & & x = \gamma\backslash(\delta', \delta) \\
 & \equiv & \qquad \text{Charn} \\
 & & \gamma ; x = (\delta', \delta) \\
 & \equiv & \qquad \text{property natural transformations, } \omega \text{ and } \mathit{0}, S \\
 & & (\gamma ; x)\mathit{0} = (\delta', \delta)\mathit{0} \quad \wedge \quad (\gamma ; x)S = (\delta', \delta)S \\
 & \equiv & \qquad \text{calculus for natural transformations, defn } (\delta', \delta) \\
 & & \gamma 0 ; x = \delta' \quad \wedge \quad \gamma S ; x = \delta \\
(*) & \equiv & \qquad \textbf{define } \delta' \textbf{ below} \text{ such that } \gamma 0 ; x = \delta' \text{ for any } x \ (\text{using } \gamma S ; x = \delta) \\
 & & \gamma S ; x = \delta.
\end{array}
$$

In order to define δ' satisfying the requirement derived at step $(*)$, we calculate

$$
\begin{array}{ll}
 & \gamma 0 ; x \\
= & \qquad \{\text{anticipating next steps, introduce an identity}\} \\
 & \gamma 0 ; K_c(0 \leq 1) ; x \\
= & \qquad \text{naturality } \gamma \text{ (“commutativity of the triangle”)} \\
 & D(0 \leq 1) ; \gamma 1 ; x \\
= & \qquad \text{using } \gamma S ; x = \delta \\
 & D(0 \leq 1) ; \delta 0
\end{array}
$$

so that we can fulfil the requirement $\gamma 0 ; x = \delta'$ by defining $\delta' = D(0 \leq 1) ; \delta 0$.

Part 3 (Proof of ‘α is initial in $\mathrm{Alg}(F)$’ where $\alpha = F\gamma\backslash\gamma S$.) Let $\varphi : Fa \to a$ be arbitrary. We have to construct a morphism $(\!|\varphi|\!) : c \to a$ in $\mathbf{C}$ such that

$$
(\heartsuit) \quad F\gamma\backslash\gamma S ; x = Fx ; \varphi \quad \equiv \quad x = (\!|\varphi|\!) .
$$

Our guess is that the required morphism $(\!|\varphi|\!)$ can be written as $\gamma\backslash\alpha$ for some suitably chosen D-cocone α. Based on type considerations one may derive a definition of α. We omit the derivation and just state the result: $\alpha 0 = (\!|a|\!)$ and $\alpha S = F\alpha ; \varphi$. It remains to show that $\gamma\backslash\alpha$ for $(\!|\varphi|\!)$ fulfils the required equivalence $(\heartsuit)$. First we show the $\Leftarrow$ part:

$$
\begin{array}{ll}
 & F\gamma\backslash\gamma S ; \gamma\backslash\alpha \\
= & \qquad \text{Fusion} \\
 & F\gamma\backslash(\gamma S ; \gamma\backslash\alpha) \\
= & \qquad \text{observed just before the colimit laws} \\
 & F\gamma\backslash\alpha S \\
= & \qquad \text{definition } \alpha S = F\alpha ; \varphi
\end{array}
$$

$$
\begin{array}{ll}
 & F\gamma\backslash(F\alpha;\varphi) \\
= & \qquad \text{Fusion} \\
 & F\gamma\backslash F\alpha;\varphi \\
= & \qquad \text{functor } F \text{ preserves colimits} \\
 & F(\gamma\backslash\alpha);\varphi
\end{array}
$$

as desired. Next we show the $\Rightarrow$ part.

$$
\begin{array}{ll}
 & x = \gamma\backslash\alpha \\
\equiv & \qquad \text{Charn} \\
 & \gamma;x = \alpha \\
\Leftarrow & \qquad \text{induction} \\
 & (\gamma;x)0 = \alpha 0 \quad \wedge \quad \forall(n :: \quad (\gamma;x)n = \alpha n \ \Rightarrow \ (\gamma;x)Sn = \alpha Sn) \\
\equiv & \qquad \text{proved below in (i) and (ii)} \\
 & \mathsf{true}.
\end{array}
$$

For (i) we calculate

$$
\begin{array}{ll}
 & \gamma 0;x \\
= & \qquad \text{Charn, using } \gamma 0 : \mathbf{0} \to c \\
 & (\!|c|\!);x \\
= & \qquad \text{Fusion, using } x : c \to a \\
 & (\!|a|\!) \\
= & \qquad \text{defn } \alpha 0 \\
 & \alpha 0
\end{array}
$$

as desired. And for (ii) we calculate, assuming $(\gamma;x)n = \alpha n$,

$$
\begin{array}{ll}
 & (\gamma S;x)n \\
= & \qquad \{\text{aiming at using the premise of } (\heartsuit)\} \text{ Self} \\
 & (F\gamma;F\gamma\backslash\gamma S;x)n \\
= & \qquad \text{premise, see } (\heartsuit) \\
 & (F\gamma;Fx;\varphi)n \\
= & \qquad \text{functor} \\
 & (F(\gamma;x);\varphi)n \\
= & \qquad \text{hypothesis } (\gamma;x)n = \alpha n \\
 & (F\alpha;\varphi)n \\
= & \qquad \text{defn } \alpha \\
 & (\alpha S)n
\end{array}
$$

as desired. This completes the entire construction and proof.

Acknowledgement This note is heavily influenced by what other people have taught me in the past two years. In particular Roland Backhouse, Grant Malcolm, Lambert Meertens and Jaap van der Woude may recognise their ideas and suggestions.

References

[1] M.M. Fokkinga and E. Meijer. Program calculation properties of continuous algebras. December 1990. CWI, Amsterdam.