Monad Decomposition

Roland C. Backhouse

April 16, 1993

Chapters 7 and 8 of the 1992 STOP proceedings are devoted to the study of closure operators. At the end of chapter 8 a research programme was outlined, a goal of which was to be the derivation of programs on lists as by-products of calculations with the Kleene star. This note makes a start on that research programme by showing how calculations with the closure star (discussed in chapter 7 of the STOP proceedings) can be "lifted" to constructive proofs about monads. In particular, the fundamental closure-decomposition rule is "lifted" to a monad decomposition theorem.

1 Natural Simulations

Lattice theory is a theory about preorders. We want to view category theory as *constructive* lattice theory in which the objects of interest are relators and the preordering between them expresses a notion of simulation. Supposing that F and G are relators and γ is a spec, the objective is thus to develop a theory about terms of the form

$$\gamma \in F \gtrsim G,$$

the intended meaning of such a term being that γ is a program that enables all computations over G-structures to be simulated by computations over F-structures. In short, γ *witnesses* the fact that F *simulates* G.

In order that such a theory should truly merit the attribute "constructive" it should be the case that the witnesses to simulations are by-products of calculations in lattice theory. In other words, if we write $F \gtrsim G$ as an abbreviation of $\exists(\gamma :: \gamma \in F \gtrsim G)$ then it should be possible to abbreviate all calculations that construct a witness to a simulation to ones omitting the construction of the witness that are valid calculations in lattice theory itself.

There are several possibilities for the notion of natural simulation in the spec calculus. For concreteness we assume the following definition throughout this paper.

Definition 1 (Natural Simulation) Let F and G be relators and let γ be a spec. We say that γ witnesses the fact that F simulates G iff

a. $\gamma \in F \dot{\lessgtr} G$
(i.e. $\forall(R :: F.R \circ \gamma = \gamma \circ G.R)$),

b. $\gamma> \ = \ G.I$,

c. γ is an imp.

□

Note that is immediate from a. and b. that $\gamma< \sqsubseteq F.I$ and, hence, $F.I \circ \gamma = \gamma$.

We shall say that F *simulates* G if a γ exists such that $\gamma \in F \gtrsim G$.

Theorem 2 The simulates relation is a constructive preordering on relators. Specifically, simulates is constructively reflexive:

$$F.I \in F \gtrsim F$$

and constructively transitive:

$$\gamma \in F \gtrsim G \wedge \delta \in G \gtrsim H \ \Rightarrow \ \gamma \circ \delta \in F \gtrsim H \quad .$$

Moreover, composition of relators is constructively monotonic in both its arguments. Specifically,

$$\gamma \in F \gtrsim G \wedge \delta \in H \gtrsim K \ \Rightarrow \ \gamma \circ G.\delta \in F \bullet H \gtrsim G \bullet K \quad .$$

□

The proof of this theorem is straightforward and hence omitted. In proof hints the adjective "constructively" will be omitted.

It is important to note that the combination of $F \gtrsim G$ and $G \gtrsim F$ does *not* imply that F and G are naturally isomorphic. Nevertheless, in "lifting" lattice-theoretic notions to categorical notions we shall typically add extra requirements with the purpose of indeed deducing natural isomorphism from a combination of two simulations.

2 Monads

In category theory closure operators are cited as instances of monads. We want to take the opposite view: monads are constructive closure operators.

One definition of a closure operator is a monotonic endofunction, f, on a preorder that is both reflexive and idempotent. A first stab at a definition of a monad is thus a relator F that is constructively reflexive and constructively idempotent. A monad is thus a relator F such that there exist two specs unit and mul with

(3) $\quad \mathsf{unit} \;\in\; F \gtrsim I\!I \quad ,$

(4) $\quad \mathsf{mul} \;\in\; F \gtrsim F \bullet F \quad .$

(In category theory the symbols η and μ are normally used for unit and mul. The use of μ might, however, cause some confusion with least fixed points later on.)

Note that the combination of these two postulates with the monotonicity of composition means that monad F simulates F^i for all $i \geq 0$ (where, by definition, $F^0 = I\!I$ and $F^{i+1} = F \bullet F^i$). For example, the calculation

$$
\begin{array}{ll}
 & F \gtrsim F \bullet F \bullet F \\
\Leftarrow & \{ \text{ transitivity } \} \\
 & F \gtrsim F \bullet F \;\wedge\; F \bullet F \gtrsim F \bullet F \bullet F \\
\Leftarrow & \{ \text{ monotonicity } \} \\
 & F \gtrsim F \bullet F \;\wedge\; F \gtrsim F \bullet F \;\wedge\; F \gtrsim F
\end{array}
$$

can be lifted to the construction of a simulation as follows:

$$\begin{array}{ll} & \alpha \in F \gtrsim F \bullet F \bullet F \\ \Leftarrow & \{ \alpha \ = \ \beta \circ \gamma \} \\ & \beta \in F \gtrsim F \bullet F \ \wedge \ \gamma \in F \bullet F \gtrsim F \bullet F \bullet F \\ \Leftarrow & \{ \gamma \ = \ \delta \circ F.F.\eta \} \\ & \beta \in F \gtrsim F \bullet F \ \wedge \ \delta \in F \gtrsim F \bullet F \ \wedge \ \eta \in F \gtrsim F \quad . \end{array}$$

Thus assuming (3) and (4)

$$\mathsf{mul} \circ \mathsf{mul} \circ F.F.F.I \in F \gtrsim F \bullet F \bullet F \quad .$$

Simplifying, (since $\mathsf{mul} \circ F.F.F.I \ = \ F.F.I \circ \mathsf{mul}$ and $\mathsf{mul} \circ F.F.I \ = \ \mathsf{mul}$)

$$\mathsf{mul} \circ \mathsf{mul} \in F \gtrsim F \bullet F \bullet F \quad .$$

This however is not the only way to construct a witness. Note that in the second step $F \bullet F \bullet F$ was parsed as $(F \bullet F) \bullet F$ in order to apply the monotonicity rule. Had we parsed it as $F \bullet (F \bullet F)$ then we would have deduced

$$\mathsf{mul} \circ F.\mathsf{mul} \in F \gtrsim F \bullet F \bullet F \quad .$$

Similarly there are three obvious ways of constructing a simulation of F by itself. First, $F.I$ witnesses such a simulation. Second, since $F = I\!I \bullet F$, $\mathsf{mul} \circ \mathsf{unit}$ witnesses such a simulation. Third, since $F = F \bullet I\!I$, $\mathsf{mul} \circ F.\mathsf{unit}$ witnesses the simulation. In order to have any hope of deducing from $F \gtrsim G$ and $G \gtrsim F$ that F and G are isomorphic monads we had better require that all constructions of simulations of F^i by F lead, for each i, to the same witness. This results in the following definition of a monad.

Definition 5 (Monad) A monad is a triple $(F, \mathsf{unit}, \mathsf{mul})$ such that F is a relator and unit and mul are two specs with

a. $\mathsf{unit} \in F \gtrsim I\!I$,

b. $\mathsf{mul} \in F \gtrsim F \bullet F$,

c. $\mathsf{mul} \circ \mathsf{mul} \ = \ \mathsf{mul} \circ F.\mathsf{mul}$,

d. $\mathsf{mul} \circ F.\mathsf{unit} \quad = \quad F.I \quad = \quad \mathsf{mul} \circ \mathsf{unit} \quad .$

□

Property c. is called the *associative* law and d. the *unit* law. Together they guarantee unicity of simulations of F^i by F for each i.

Having identified a class of relators the first test that it should undergo is whether or not it is closed under natural isomorphisms. For monads we have:

Theorem 6 Suppose $(F, \mathsf{unit}_F, \mathsf{mul}_F)$ is a monad and γ is a bijection with $\gamma< = F.I$. Then $(F^\gamma, \gamma\cup \circ \mathsf{unit}_F, \gamma\cup \circ \mathsf{mul}_F \circ F.\gamma \circ \gamma)$ is a monad where $F^\gamma.R \quad = \quad \gamma\cup \circ F.R \circ \gamma$ for all R.

Proof By definition, $\gamma \in F \cong F^\gamma$ and hence $\gamma \in F \gtrsim F^\gamma$ and $\gamma\cup \in F^\gamma \gtrsim F$. The construction of unit is a straightforward application of transitivity. The construction of mul is obtained by lifting the following calculation:

$$F^\gamma \gtrsim F^\gamma \bullet F^\gamma$$

$$\Leftarrow \qquad \{ \text{ transitivity and monotonicity } \}$$

$$F^\gamma \gtrsim F \bullet F \wedge F \gtrsim F^\gamma$$

$$\Leftarrow \qquad \{ \text{ transitivity } \}$$

$$F^\gamma \gtrsim F \wedge F \gtrsim F \bullet F \wedge F \gtrsim F^\gamma \quad .$$

This leads to the witness

$$\gamma\cup \circ \mathsf{mul}_F \circ \gamma \circ F^\gamma.\gamma$$

which equals

$$\gamma\cup \circ \mathsf{mul}_F \circ F.\gamma \circ \gamma \quad .$$

Verification of 5c. and 5d. is tedious but straightforward.
□

3 A Class of Monads

A class of monads is introduced in sections 9.4 and 9.5 of (part 2 of) the 1992 STOP proceedings. In this section a short summary is given of the definition and properties of this class. The notation used here is somewhat different to that in the proceedings.

Let F be an arbitrary relator. Define the binary relator $\oplus_F$ by

$$X \oplus_F Y \quad = \quad X + F.Y \qquad .$$

Define the unary relator $F^{\text{✡}}$ by

$$F^{\text{✡}}.X \quad = \quad \mu(X\oplus_F) \qquad .$$

Define in addition the specs η_F and τ_F by

$$(7) \quad \eta_F \quad = \quad \hookleftarrow \circ F.F^{\text{✡}}.I \;=\; F^{\text{✡}}.I \circ \hookleftarrow$$

$$(8) \quad \tau_F \quad = \quad \hookrightarrow \;=\; F^{\text{✡}}.I \circ \hookrightarrow$$

Finally, denote $(\![\, I\oplus_F; R \,]\!)$ by $(\![\, R \,]\!)_F$. Then we have

(9) $\quad \tau_F \in F^{\text{✡}} \gtrsim \mathbb{I} \quad ,$

(10) $\quad \eta_F \in F^{\text{✡}} \gtrsim F \bullet F^{\text{✡}} \quad ,$

(11) $\quad (F^{\text{✡}}, \tau_F, (\![\, F^{\text{✡}}.I \triangledown \eta_F \,]\!)_F)$ is a monad.

(Properties (9) and (10) are 283(a) and 283(b) in the 1992 proceedings, (11) is corollary 288: make the substitutions $\varpi := F^{\text{✡}}, F := I\oplus, H := F, /R \circ \varpi S = (\![\, S \triangledown R \,]\!)_F$. Note that $(\![\, F^{\text{✡}}.I \triangledown \eta_F \,]\!)_F$ has been taken as the third component of the monad rather than $(\![\, I \triangledown \eta_F \,]\!)_F$ so that part (e) of corollary 288 is brought into line with the definition of monad given here.)

The use of the closure star here is meant to link in the reader's mind the monadic relator $F^{\text{✡}}$ with the closure operator $f^{\text{✡}}$ studied in chapter 7 of the 1992 proceedings. Indeed the goal of this paper is to lift a number of the properties of closure operators to constructions of simulations and isomorphisms between relators defined using the $^{\text{✡}}$ operator. We shall be particularly concerned with constructing an isomorphism between the relators $(F + G)^{\text{✡}}$ and $G^{\text{✡}} \bullet (F \bullet G^{\text{✡}})^{\text{✡}}$ — in other words establishing that the closure decomposition theorem is constructively valid. The theorem we obtain is called the monad decomposition theorem and it is this theorem that gives the paper its title.

First, however, let us give some elementary examples of monads together with some hints on how to read Wadler's Marktoberdorf lecture notes and/or John Hughes STOP contribution in combination with this paper.

The simplest examples are obtained by taking F to be a constant relator. Suppose A is a monotype, and let $\hat{A}$ denote the relator $X \mapsto A$. Then

$$
\begin{array}{ll}
 & \hat{A}^{\star} \\
= & \{\ F^{\star}.R = \mu(Y \mapsto R + F.Y) \ = \ R + F.F^{\star}.R, \\
 & \quad \text{i.e. } F^{\star} \ = \ I\!I + F \bullet F^{\star}\ \} \\
 & I\!I + \hat{A} \bullet \hat{A}^{\star} \\
= & \{\ \hat{A} \bullet F \ = \ \hat{A}\ \} \\
 & I\!I + \hat{A} \quad .
\end{array}
$$

Taking A to be $\perp\!\!\perp$ we have in particular that $\hat{\perp\!\!\perp}^{\star} = I\!I + \perp\!\!\perp$. Since $I\!I + \perp\!\!\perp$ is isomorphic to $I\!I$ it follows that the identity relator is a monad (as might be expected!). This is Wadler's first example. Hughes first example – "Maybe" – is obtained by taking A to be $1\!\!1$ (the unit type). Wadler's second example is obtained by substituting *String* for A – he calls it "Exception."

From the above the unit of $\hat{A}^{\star}$ is $\hookrightarrow$ and the multiplier is $\hookrightarrow \triangledown I$. Hughes and Wadler define monads not in terms of the unit, multiplier and relator but rather in terms of the unit and a function called "bind" by Hughes. This function (as indeed explained by Wadler) is easily defined in terms of the unit and multiplier. Specifically, for arbitrary monad F,

$$\mathsf{bind}_F.R \quad = \quad \mathsf{mul}_F \circ F.R \quad .$$

(For function f, $\mathsf{bind}_F.f$ is the function $a \mapsto a$ "bind"f in Hughes' notation.) In particular, for $M = \hat{A}^{\star}$

$$\mathsf{bind}_M.R \quad = \quad (\hookrightarrow \circ R) \triangledown A \quad .$$

4 The Decomposition Theorem – 1st Proof

As discussed in the 1992 proceedings, one of the most important properties of the * operator (at the lattice level) is the closure decomposition theorem. It is this theorem that is at the heart of facts like the relationship between equivalence closure and symmetric and reflexive-transitive closure (namely the former is the forwards composition of the latter two) and the relationship between reflexive-transitive closure and reflexive closure and transitive closure. A *monad* decomposition theorem is likely to be equally fundamental although I have not been able to find one in the category-theory literature.

In fact, the monad decomposition theorem has a very straightforward proof in the spec calculus as we will now demonstrate. Unfortunately, however, this proof is likely to be rejected by functional programmers for reasons to be explained after the proof has been given. In a later section, therefore, we give a rather more complicated proof of the theorem which should be acceptable to users of conventional programming languages.

Theorem 12 (Monad Decomposition) For all relators F and G,

$$(F+G)^{\star} \;\cong\; G^{\star} \bullet (F \bullet G^{\star})^{\star} \quad .$$

Proof We first observe the equality:

$$G^{\star}.(F \bullet G^{\star})^{\star}.R \quad = \quad \mu(X \mapsto (R + F.X) + G.X)$$

for all specs R.

$$G^{\star}.(F \bullet G^{\star})^{\star}.R$$

$$= \qquad \{ \text{ definition of } ^{\star} \; \}$$

$$G^{\star}.\mu(X \mapsto R + F.G^{\star}.X)$$

$$= \qquad \{ \; \mu\text{-fusion: } f.\mu(g \bullet f) \;=\; \mu(f \bullet g), \quad f := G^{\star}, g := X \mapsto R + F.X \; \}$$

$$\mu(X \mapsto G^{\star}.(R + F.X))$$

$$= \qquad \{ \text{ definition of } ^{\star} \; \}$$

$$\mu(X \mapsto \mu(Y \mapsto (R + F.X) + G.Y))$$

$$= \quad \{ \text{ diagonal rule: } \mu(X \mapsto \mu(Y \mapsto X \oplus Y)) \;=\; \mu(X \mapsto X \oplus X) \}$$

$$\mu(X \mapsto (R + F.X) + G.X) \quad .$$

The proof of the decomposition theorem is now a straightforward application of theorem 220 in part 2 of the 1992 proceedings:

$$G^{\star} \bullet (F \bullet G^{\star})^{\star}$$

$$= \quad \{ \text{ above } \}$$

$$(R \mapsto \mu(X \mapsto (R + F.X) + G.X))$$

$$\cong \quad \{ \text{ Theorem 220: }$$
$$(\hookrightarrow \circ \hookrightarrow) \triangledown (\hookleftarrow + I) \in \oplus \cong \otimes$$
$$\text{where } X \oplus Y \;=\; (X + F.Y) + G.Y$$
$$\text{and } X \otimes Y \;=\; X + (F.Y + G.Y)$$
$$\text{Hence } (\![I \otimes ; (\hookrightarrow \circ \hookrightarrow) \triangledown (\hookleftarrow + I)]\!) \text{ is the required isomorphism } \}$$

$$(R \mapsto \mu(X \mapsto R + (F.X + G.X)))$$

$$= \quad \{ \text{ definition of } {}^{\star} \}$$

$$(F + G)^{\star} \quad .$$

Thus we have constructed

$$(\![I \otimes ;\ (\hookrightarrow \circ \hookrightarrow) \triangledown (\hookleftarrow + I)]\!) \in G^{\star} \bullet (F \bullet G^{\star})^{\star} \cong (F + G)^{\star} \quad .$$

Its inverse (by theorem 220) is

$$(\![I \oplus ;\ (I + \hookrightarrow) \triangledown (\hookleftarrow \circ \hookleftarrow)]\!) \in (F + G)^{\star} \cong G^{\star} \bullet (F \bullet G^{\star})^{\star} \quad .$$

Both $\oplus$ and $\otimes$ are defined in the hint above.

□

5 The Decomposition Theorem – 2nd Proof

The simplicity and straightforwardness of the proof of monad decomposition is evidence of the conciseness and elegance of the spec calculus. The "real" functional programmer is likely, however, to throw up his hands in horror. "But where is the program?" he is likely to demand.

The reason that the "real" functional programmer is not able to see the program is that "modern" functional programming languages demand the invention of "meaningful" identifiers for the datatype constructors, and much of functional programming boils down to replacing one identifier by another. Let us illustrate this with an instance of the monad decomposition theorem expressed in the language Haskell.

Suppose we define four datatypes as follows:

```
data M a  =  Om a | Two (M a) (M a) | Three (M a) (M a) (M a)
data N a  =  On a | Drie (N a) (N a) (N a)
data P a  =  Twee (N a) (N a)
data Q a  =  Oq a | P (Q a)
```

("Twee" and "drie" are the Dutch names for "two" and "three".) In terms of the spec calculus,

$$\begin{array}{rcl} M & = & (Sq + Cube)^{\star} \\ N & = & Cube^{\star} \\ P & = & Sq \bullet N \\ Q & = & P^{\star} \quad , \end{array}$$

$$\begin{array}{rl} \text{where} & Cube.R \;=\; R \times R \times R \\ \text{and} & Sq.R \;=\; R \times R \quad . \end{array}$$

Thus by the monad decomposition theorem,

$$M \cong N \bullet Q \quad .$$

An element of type M is depicted in fig.1. The corresponding element of type $N \bullet Q$ predicted by the isomorphism we are about to construct is shown in fig.2. The construction of one tree from the other is very straightforward: replace the constructor "`Om`" everywhere by "`On Oq`", the constructor "`Two`"

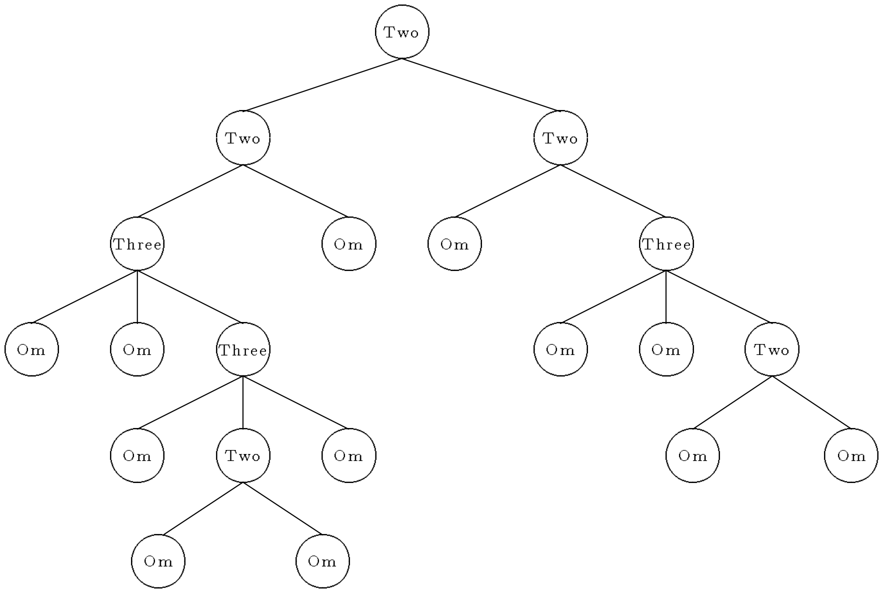

Figure 1: Element of Type M

by "`On Twee`"and the constructor "`Three`" by "`Drie`" . (Important note: these figures were constructed *after* I had derived the second proof of the monad decomposition theorem and not before.) In this way one can (literally) see the "real" program witnessing the isomorphism between M and $N \bullet Q$.

In this section an alternative derivation of the monad decomposition theorem is given. Two specs are constructed witnessing the claimed isomorphism. The specs constructed here are equal to the specs constructed in the previous section, but, unlike those specs, the ones constructed here are directly translatable into programs written in a functional programming language. (This translation has indeed been carried out for the particular datatypes defined above; the resulting program is included in the appendix. It would, of course, have been far better to carry out the translation for arbitrary datatypes F and G rather than for specific ones but I do not know of any programming language that permits programs to be parameterised on datatypes.)

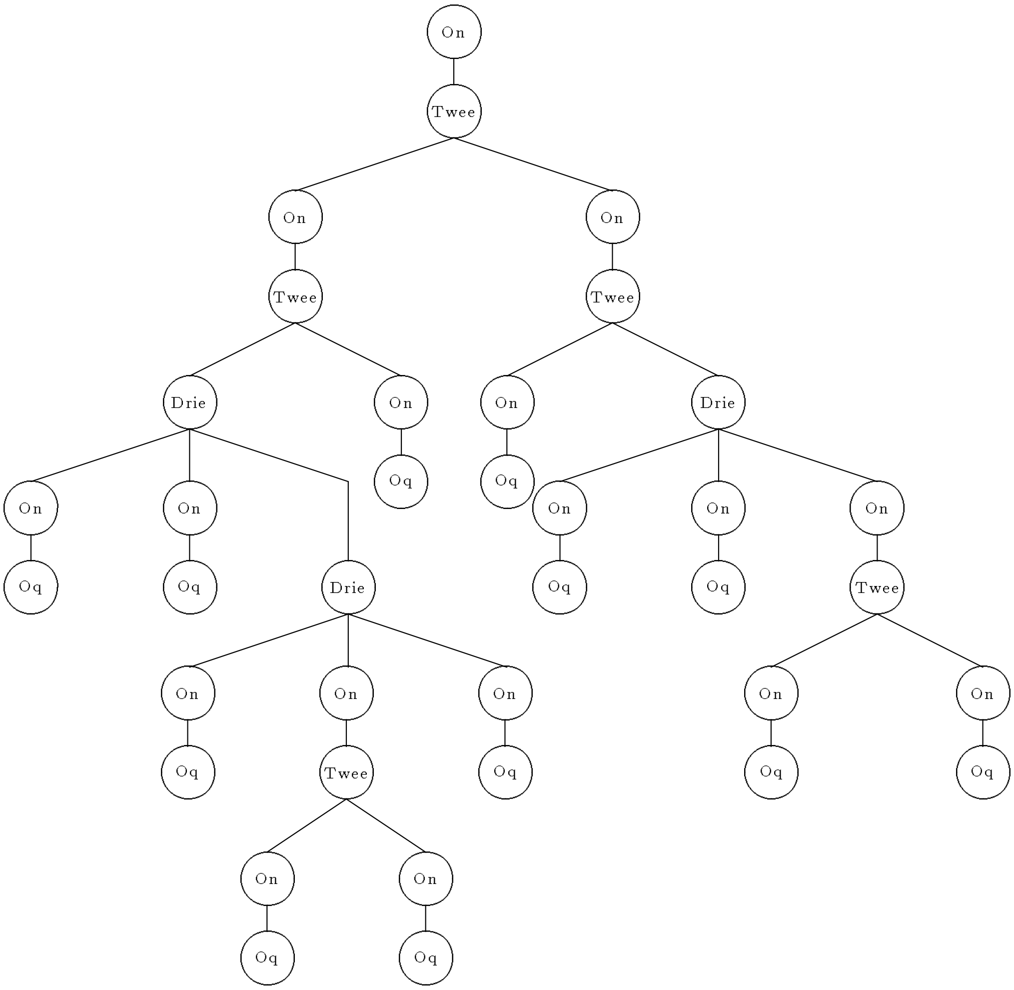

Figure 2: Element of Type $N \bullet Q$

The strategy is to augment the calculations in chapter 7 with the construction of witnesses to the relevant simulations. The properties needed to do this are properties (9) and (10) given earlier (repeated here for the sake of convenience)

(13) $\tau_F \quad \in \quad F^{\star} \gtrsim I\!I \quad ,$

(14) $\eta_F \quad \in \quad F^{\star} \gtrsim F \bullet F^{\star} \quad ,$

the decomposition property of disjoint sum

(15) $\alpha \triangledown \beta \in F \gtrsim G + H \;\Leftarrow\; \alpha \in F \gtrsim G \wedge \beta \in F \gtrsim H \quad ,$

and its inverse

(16) $\alpha \circ I + I \in F \gtrsim G + H \;\Rightarrow\; \alpha \circ \hookrightarrow \in F \gtrsim G \;\wedge\; \alpha \circ \hookleftarrow \in F \gtrsim H \quad ,$

and the extremality property of $F^{\star}$

(17) $(\![\, \alpha \,]\!)_F \in H \gtrsim F^{\star} \bullet G \;\Leftarrow\; \alpha \circ I + F.H.I \in H \gtrsim G + F \bullet H \quad .$

Properties (13) – (17) are universally quantified over specs α and β and relators F, G and H. In addition we make extensive use of the fact that the simulates relation is a constructive preordering on relators and composition of relators is constructively monotonic in both its arguments.

The proofs of (15), (16) and (17) are given in the appendix. The combination of (15), (16) and the properties

$$\alpha \triangledown \beta \circ \hookrightarrow \;=\; \alpha \quad , \alpha \triangledown \beta \circ \hookleftarrow \;=\; \beta \quad ,$$

$$(\alpha \circ \hookrightarrow) \triangledown (\alpha \circ \hookleftarrow) \;=\; \alpha \circ I + I$$

forms an instance of a "constructive Galois connection", or "adjunction" in the category theory terminology. (See "Adjunctions in the spec calculus", Ameland proceedings, 1991.)

Note also that (16) and (17) permit a certain flexibility in the choice of simulations. We shall exploit this flexibility to construct simulations that correspond to those that a functional programmer would construct rather than the more concise ones we normally construct in the spec calculus. The flexibility is syntactic not semantic. That is, we have by the domain trading rule for catamorphisms

$$
\begin{array}{llcl}
(18) & (\![R]\!)_F & = & (\![S]\!)_F \in H \gtrsim F^{\star} \bullet G \\
& \Leftarrow \quad R \circ I + F.H.I & = & S \circ I + F.H.I \in H \gtrsim G + F \bullet H \ .
\end{array}
$$

Likewise we have

$$
\begin{array}{llcl}
(19) & R \circ I + I & = & S \circ I + I \\
& \Rightarrow \quad (R \circ \hookrightarrow \ = \ S \circ \hookrightarrow) \wedge (R \circ \hookleftarrow \ = \ S \circ \hookleftarrow) & & .
\end{array}
$$

Thus the flexibility in use of the rules (16) and (17) does not destroy any unicity properties of the obtained simulations.

Assume that F and G are arbitrary relators. For brevity define the relators H and K by

$$
\begin{array}{llcl}
(20) & H & = & G^{\star} \bullet (F \bullet G^{\star})^{\star} \\
(21) & K & = & F \bullet G^{\star}
\end{array}
$$

We begin by showing that H simulates $(F + G)^{\star}$.

$$
\begin{array}{ll}
& R_0 \in H \gtrsim (F + G)^{\star} \\
\Leftarrow & \{ \ R_0 \ = \ (\![X]\!)_{F+G}, (17) \ \} \\
& X \circ I + (F + G).H.I \in H \gtrsim \mathbb{I} + (F + G) \bullet H \\
\Leftarrow & \{ \ (15), \quad (F + G) \bullet H \ = \ (F \bullet H) + (G \bullet H) \ , \} \\
& \{ \ X \ = \ R_1 \triangledown (R_2 \triangledown R_3) \ \} \\
& R_1 \in H \gtrsim \mathbb{I} \wedge R_2 \circ F.H.I \in H \gtrsim F \bullet H \wedge R_3 \circ G.H.I \in H \gtrsim G \bullet H \ .
\end{array}
$$

This leaves us with the task of constructing three simulations.

For R_1 we calculate:

$$
\begin{array}{ll}
& R_1 \in H \gtrsim \mathbb{I} \\
\Leftarrow & \{ \ \text{definition (20) of } H, \ R_1 \ = \ X \circ Y \ \} \\
& X \in G^{\star} \gtrsim \mathbb{I} \ \wedge \ Y \in (F \bullet G^{\star})^{\star} \gtrsim \mathbb{I}
\end{array}
$$

$$\Leftarrow \quad \{ \text{(13), definition of } K \ \}$$

$$X \ = \ \tau_G \ \ \wedge \ \ Y \ = \ \tau_K$$

The strategy for calculating R_2 and R_3 is first to ignore the flexibility afforded by the domain restrictions "$\circ\, F.H.I$" and "$\circ\, G.H.I$". We calculate

$$X \in H \gtrsim F \bullet H$$

$$\equiv \quad \{ \text{ definition of } H \text{ and } K \ \}$$

$$X \in G^{\star} \bullet K^{\star} \gtrsim K \bullet K^{\star}$$

$$\Leftarrow \quad \{ \text{ monotonicity, } X \ = \ Y \circ Z \ \}$$

$$Y \in G^{\star} \gtrsim I\!I \ \wedge \ Z \in K^{\star} \gtrsim K \bullet K^{\star}$$

$$\Leftarrow \quad \{ \text{ (13) and (14) } \}$$

$$Y \ = \ \tau_G \ \wedge \ Z \ = \ \eta_K$$

Thus, we have constructed

$$\tau_G \circ \eta_K \in H \gtrsim F \bullet H.$$

Now we exploit the flexibility. We note that

$$\tau_G \circ \eta_K$$

$$= \quad \{ \text{ definitions: (7) and (8) } \}$$

$$\hookrightarrow \circ \hookleftarrow \circ K.K^{\star}.I$$

$$= \quad \{ \text{ definitions of } H \text{ and } K \ \}$$

$$\hookrightarrow \circ \hookleftarrow \circ F.H.I$$

Thus for R_2 there are two possibilities

(22) $R_2 \ = \ \tau_G \circ \eta_K$

(23) $R_2 \ = \ \hookrightarrow \circ \hookleftarrow$.

In a similar way – the details are left to the reader – we construct

(24) $\eta_G \circ G.H.I \in H \gtrsim G \bullet H.$

This gives three reasonable choices for R_3, either

$$\begin{array}{llll} (25) & R_3 & = & \eta_G \circ G.H.I, \quad \text{or} \\ (26) & R_3 & = & \eta_G, \quad \text{or} \\ (27) & R_3 & = & \hookleftarrow . \end{array}$$

The conclusion of this calculation is that, with the definition

$$(28) \quad R_0 \quad = \quad (\!|\, (\tau_G \circ \tau_K) \triangledown ((\tau_G \circ \eta_K) \triangledown \eta_G) \,|\!)_{F+G} \qquad ,$$

we have

$$(29) \quad R_0 \in H \gtrsim (F+G)^{\star}$$

and

$$(30) \quad R_0 \quad = \quad (\!|\, (\hookrightarrow \circ \hookrightarrow) \triangledown ((\hookrightarrow \circ \hookleftarrow) \triangledown \hookleftarrow) \,|\!)_{F+G} \qquad .$$

(Note that the right side of (30) was predicted by the first proof of the monad decomposition theorem.)

For the other half of the isomorphism we first have to construct a spec witnessing the fact that $(F+G)^{\star}$ simulates H. We will not give the details of the calculation but just show the result. The first step is to construct a witness ignoring the flexibility in (17) and (16). We obtain that with S_0 – S_4 defined by

$$\begin{array}{llll} (31) & S_0 & = & (\!|\, S_1 \triangledown S_2 \,|\!)_G \\ (32) & S_1 & = & (\!|\, \tau_{F+G} \triangledown S_3 \,|\!)_K \\ (33) & S_2 & = & \eta_{F+G} \circ \hookleftarrow \circ\, G.(F+G)^{\star}.I \\ (34) & S_3 & = & \eta_{F+G} \circ \hookrightarrow \circ\, F.S_4 \\ (35) & S_4 & = & (\!|\, (F+G)^{\star}.I \triangledown S_2 \,|\!)_G \end{array}$$

then we have

$$\begin{array}{llll} (36) & S_0 & \in & (F+G)^{\star} \gtrsim H \\ (37) & S_1 & \in & (F+G)^{\star} \gtrsim K^{\star} \\ (38) & S_2 & \in & (F+G)^{\star} \gtrsim G \bullet (F+G)^{\star} \\ (39) & S_3 & \in & (F+G)^{\star} \gtrsim K \bullet (F+G)^{\star} \\ (40) & S_4 & \in & (F+G)^{\star} \gtrsim G^{\star} \bullet (F+G)^{\star} \qquad . \end{array}$$

Note that the correctness of (36) – (40) can be verified by the well-known type inference algorithm due to Milner.

The domain restrictions in the definitions (31) – (35) can be removed in two stages. First the explicit domain restrictions in the formulae for S_2 and S_4 can be removed in order to obtain the programs that a functional programmer would construct. Second the implicit domain restrictions in η_{F+G} can be removed to obtain the program that the spec-calculist would construct.

The domain restriction in S_2 is soon dismissed: by the computation rule for disjoint sum and the type of η_{F+G} we have

$$S_2 \quad = \quad \eta_{F+G} \circ \hookleftarrow \quad .$$

The domain restriction in S_4 cannot be removed in S_4 itself (without violating (40)). It can be removed from S_1 as follows:

$$\begin{array}{ll} & S_3 \\ = & \{ \text{ definition } \} \\ & \eta_{F+G} \circ \hookrightarrow \circ F.S_4 \\ = & \{ \; S_4 \; = \; (\!|\, I \triangledown S_2 \,|\!)_G \circ G^{\ast}.(F+G)^{\ast}.I \text{ by map fusion } \} \\ & \eta_{F+G} \circ \hookrightarrow \circ F.(\!|\, I \triangledown S_2 \,|\!)_G \circ K.(F+G)^{\ast}.I \quad . \end{array}$$

Thus by (18)

$$S_1 \quad = \quad (\!|\, \tau_{F+G} \triangledown (\eta_{F+G} \circ \hookrightarrow \circ F.(\!|\, I \triangledown S_2 \,|\!)_G) \,|\!)_K \quad .$$

We conclude that with

$$\begin{array}{llll} (41) & T_0 & = & (\!|\, T_1 \triangledown T_2 \,|\!)_G \\ (42) & T_1 & = & (\!|\, \tau_{F+G} \triangledown T_3 \,|\!)_K \\ (43) & T_2 & = & \eta_{F+G} \circ \hookleftarrow \\ (44) & T_3 & = & \eta_{F+G} \circ \hookrightarrow \circ F.T_4 \\ (45) & T_4 & = & (\!|\, I \triangledown T_2 \,|\!)_G \end{array}$$

we have

$$(46) \quad S_0 \quad = \quad T_0 \quad .$$

The spec T_0 is what a functional programmer would construct as the inverse of R_0. We still have to show, however, that it is indeed R_0's inverse. As preliminary to this we remove the domain restrictions implicit in the occurrences of η_{F+G}. Since

$$\begin{array}{ll}
& \eta_{F+G} \circ \hookleftarrow \\
= & \quad \{ \text{ definition } \} \\
& \hookleftarrow \circ (F+G).(F+G)^{\star}.I \circ \hookleftarrow \\
= & \quad \{ \text{ computation rule } \} \\
& \hookleftarrow \circ \hookleftarrow \circ G.(F+G)^{\star}.I
\end{array}$$

and, similarly,

$$\eta_{F+G} \circ \hookrightarrow \quad = \quad \hookleftarrow \circ \hookrightarrow \circ F.(F+G)^{\star}.I$$

we have, by (18),

$$T_0 \quad = \quad (\![\, T_1 \triangledown (\hookleftarrow \circ \hookleftarrow) \,]\!)_G$$

and

$$\begin{array}{ll}
& T_3 \\
= & \quad \{ \text{ above } \} \\
& \hookleftarrow \circ \hookrightarrow \circ F.((F+G)^{\star}.I \circ T_4) \\
= & \quad \{ T_4 \circ G^{\star}.(F+G)^{\star}.I \quad = \quad S_4 \in (F+G)^{\star} \gtrsim G^{\star} \bullet (F+G)^{\star} \ \} \\
& \hookleftarrow \circ \hookrightarrow \circ F.T_4 \quad .
\end{array}$$

Thus with the definitions:

$$\begin{array}{llll}
(47) & U_0 & = & (\![\, U_1 \triangledown (\hookleftarrow \circ \hookleftarrow) \,]\!)_G \\
(48) & U_1 & = & (\![\, \hookrightarrow \triangledown (\hookleftarrow \circ \hookrightarrow \circ F.U_4) \,]\!)_K \\
(49) & U_4 & = & (\![\, I \triangledown (\hookleftarrow \circ \hookleftarrow) \,]\!)_G
\end{array}$$

we have

(50) $U_0 \;=\; S_0 \;=\; T_0 \in (F+G)^{\text{✡}} \gtrsim H \quad .$

The spec U_0 is what the speculist would construct directly using the flexibility in rule (17). Note that it is the most concise of the three simulations.

Since we already have a formula for $R_0{}^{\cup}$ – $([\, I\oplus; (I+\hookrightarrow)\triangledown(\hookleftarrow\circ\hookleftarrow)\,])$ – the following calculation shows that $U_0 \;=\; R_0{}^{\cup}$.

$$
\begin{array}{ll}
& U_0 \;=\; ([\, I\oplus;\; (I+\hookrightarrow)\triangledown(\hookleftarrow\circ\hookleftarrow)\,]) \\
\equiv & \{\text{ uep, } U_0{>} \;=\; H.I \;=\; \mu(I\oplus) \;\} \\
& U_0 \;=\; (I+\hookrightarrow)\triangledown(\hookleftarrow\circ\hookleftarrow) \;\circ\; (I+F.U_0)+G.U_0 \\
\equiv & \{\text{ definition: (47), computation rule } \} \\
& U_1 \triangledown (\hookleftarrow\circ\hookleftarrow) \;\circ\; I+G.U_0 \\
& \;=\; (I+\hookrightarrow)\triangledown(\hookleftarrow\circ\hookleftarrow) \;\circ\; (I+F.U_0)+G.U_0 \\
\equiv & \{\text{ fusion and } \triangledown\text{-decomposition } \} \\
& U_1 \;=\; I+(\hookrightarrow\circ F.U_0) \\
\equiv & \{\text{ definition: (48), computation rule } \} \\
& \hookrightarrow\triangledown(\hookleftarrow\circ\hookrightarrow\circ F.U_4) \;\circ\; I+K.U_1 \;=\; I+(\hookrightarrow\circ F.U_0) \\
\equiv & \{\text{ fusion and } \triangledown\text{-decomposition } \} \\
& \hookrightarrow\circ F.U_4\circ K.U_1 \;=\; \hookrightarrow\circ F.U_0 \\
\Leftarrow & \{\text{ Leibniz, } K \;=\; F\bullet G^{\text{✡}} \;\} \\
& U_4\circ G^{\text{✡}}.U_1 \;=\; U_0 \\
\equiv & \{\text{ definition: (49), map fusion } \} \\
& ([\, U_1\triangledown(\hookleftarrow\circ\hookleftarrow)\,])_G \;=\; U_0 \\
\equiv & \{\text{ definition: (47) } \} \\
& \mathit{true} \quad .
\end{array}
$$

6 Conclusion

We have given two proofs of a monad decomposition theorem. The first is a short, concise calculation in the true spec-calculus style. The second is somewhat longer but results in expressions that are directly translatable into programs written in a conventional programming language.

The monad decomposition theorem proved here is just the first in a series of calculations we intend to carry out demonstrating how properties of closure operators and of regular algebra can be lifted to powerful and useful properties of datatypes.

As a final remark, note that properties c. and d. in the definition of a monad are nowhere used in the proof of the decomposition theorem.

7 Appendix

This appendix contains proofs of the various lemmas and theorems cited in the report that are not already present in the 1992 proceedings.

The first theorem is the theorem I discussed during the 1989 workshop (under the title "Naturality of Homomorphisms") and which led to all the work on the spec calculus. Somehow I forgot to include it in the first and subsequent versions of the "red book".

Theorem 51 (Naturality of Map Relators)
For all $\sim\ \in \{\leftsquigarrow, \leftrightsquigarrow, \rightsquigarrow\}$ and all specs R and S,

$$(\!|\,\_\,|\!) \ \in\ (S \sim \varpi.R) \mathrel{\dot{\leftsquigarrow}} (S \sim R \otimes S) \quad .$$

Proof Choosing $\trianglelefteq\ \in \{\sqsupseteq, =, \sqsubseteq\}$ appropriately we have

$$
\begin{array}{ll}
 & (\!|\,\_\,|\!) \in (S \sim \varpi.R) \mathrel{\dot{\leftsquigarrow}} (S \sim R \otimes S) \\
\Leftarrow & \qquad \{\ (\!|\,\_\,|\!) \text{ is a total function }\ \} \\
 & \forall(U,V ::\ S \circ (\!|\, U \,|\!) \ \trianglelefteq\ (\!|\, V \,|\!) \circ \varpi.R \\
 & \qquad\qquad\qquad \Leftarrow\quad S \circ U \ \trianglelefteq\ V \ \circ\ R \otimes S \\
 & \quad)
\end{array}
$$

But, for all R, S, U and V,

$$
\begin{array}{ll}
 & S \circ (\![\, U \,]\!) \ \trianglelefteq\ (\![\, V \,]\!) \circ \varpi.R \\
\equiv & \qquad \{ \ \text{map fusion: } 213 \ \} \\
 & S \circ (\![\, U \,]\!) \ \trianglelefteq\ (\![\, V \circ R \otimes I \,]\!) \\
\Leftarrow & \qquad \{ \ \text{catamorphism fusion: } 170 \ \} \\
 & S \ \circ\ U \ \trianglelefteq\ V \ \circ\ R \otimes I \ \circ\ I \otimes S \\
\equiv & \qquad \{ \ \otimes \text{ is a relator } \} \\
 & S \ \circ\ U \ \trianglelefteq\ V \ \circ\ R \otimes S \quad .
\end{array}
$$

□

The second theorem is more general than that which is needed for this report. It is a theorem about map relators and replaces the theorem proved on p.128 of the proceedings. (The latter proved not to be strong enough for the current purposes.) Three lemmas lead the way.

Lemma 52 $\quad (\![\, (G.I)\otimes;\ T \,]\!) \ = \ (\![\, I\otimes;\ T \ \circ\ G.I \otimes I \,]\!) \quad .$

Proof

$$
\begin{array}{ll}
 & (\![\, (G.I)\otimes;\ T \,]\!) \\
= & \qquad \{ \ \text{theorem } 218 \ \} \\
 & (\![\, I\otimes;\ T \,]\!) \ \circ\ \varpi.G.I \\
= & \qquad \{ \ \text{map fusion: theorem } 213 \ \} \\
 & (\![\, I\otimes;\ T \ \circ\ G.I \otimes I \,]\!) \quad .
\end{array}
$$

□

Lemma 53

$$
R{>} \ = \ G.I \otimes F.I \ \wedge\ R{<} \sqsubseteq F.I \ \Rightarrow\ (\![\, I\otimes;\ R \,]\!){>} \ = \ \varpi.G.I \quad .
$$

Proof Assume $R{>} \ = \ G.I \otimes F.I \ \wedge\ R{<} \sqsubseteq F.I$. Then,

$$
\begin{array}{ll}
 & (\![\, I\otimes;\ R \,]\!) \\
= & \qquad \{ \ R{>} \ = \ G.I \otimes F.I \text{ , domains } \} \\
 & (\![\, I\otimes;\ R \ \circ\ G.I \otimes I \,]\!) \\
= & \qquad \{ \ \text{lemma } 52 \ \} \\
 & (\![\, (G.I)\otimes;\ R \,]\!) \quad .
\end{array}
$$

Hence,

$$
\begin{array}{ll}
 & (\![\, I\otimes;\ R \,]\!) \;=\; \varpi.G.I \\
\Leftarrow & \qquad \{ \text{ above, totality: theorem 178 } \} \\
 & R{>} \sqsupseteq G.I \otimes R{<} \\
\Leftarrow & \qquad \{ \;\; R{>} \;=\; G.I \otimes F.I \;\; \} \\
 & F.I \sqsupseteq R{<} \quad .
\end{array}
$$

□

Corollary 54 $\quad R \in F \gtrsim G \otimes F \;\Rightarrow\; (\![\, I\otimes;\ R \,]\!){>} \;=\; \varpi.G.I \quad .$

□

Lemma 55 For all $\dot{\sim} \;\in \{\dot{\leftsquigarrow}, \dot{\leftrightsquigarrow}, \dot{\rightsquigarrow}\}$

$$
(\![\, I\otimes;\ R \,]\!) \in F \;\dot{\sim}\; \varpi \bullet G \;\Leftarrow\; R \in F \;\dot{\sim}\; G \otimes F \quad .
$$

Proof

$$
\begin{array}{ll}
 & (\![\, I\otimes;\ R \,]\!) \in F \;\dot{\sim}\; \varpi \bullet G \\
\equiv & \qquad \{ \;\; \text{definition} \;\; \} \\
 & \forall(S ::\ (\![\, I\otimes;\ R \,]\!) \in F.S \leftrightsquigarrow \varpi.G.S) \\
\Leftarrow & \qquad \{ \;\; \text{naturality of map relators: 51} \;\; \} \\
 & \forall(S ::\ R \in F.S \leftrightsquigarrow G.S \otimes F.S) \\
\equiv & \qquad \{ \;\; \text{definition} \;\; \} \\
 & R \in F \;\dot{\sim}\; G \otimes F
\end{array}
$$

□

Theorem 56

$$
(\![\, I\otimes;\ T \,]\!) \in F \gtrsim \varpi \bullet G \;\;\Leftarrow\;\; T \circ I \otimes F.I \in F \gtrsim G \otimes F \quad .
$$

Proof By combining lemmas 54 and 55, with R instantiated to $T \circ I {\otimes} F.I$, we obtain:

$$
(\![\, I\otimes;\ T \,]\!) \in F \gtrsim \varpi \bullet G \;\Leftarrow\; T \circ I \otimes F.I \in F \gtrsim G \otimes F \quad .
$$

But, by a simple application of the domain trading rule for catamorphisms (theorem 175),

$$(\![\, I\otimes;\ T\,]\!) \ = \ (\![\, I\otimes;\ T \circ I \otimes F.I\,]\!) \quad .$$

□

Theorem 57

$$(\![\,(G.I)\otimes;\ T\,]\!) \in F \gtrsim \varpi \bullet G \ \Leftarrow\ T \circ G.I \otimes F.I \in F \gtrsim G \otimes F \quad .$$

Proof Immediate from lemma 52 together with theorem 56.

□

The monad simulation theorem is obtained directly from theorem 56 by straightforward substitution:

Theorem 58 (Monad Simulation)

$$(\![\, T\,]\!)_F \in H \gtrsim F^{\star} \bullet G \quad \Leftarrow \quad T \circ I + F.H.I \in H \gtrsim G + F{\bullet}H \quad .$$

□

With the aid of the monad simulation theorem we can verify that the relator $F^{\star}$ can indeed be furnished with a unit and multiplier making it a monad.

Lemma 59

$$\begin{array}{ll} & (\![\, F^{\star}.I \triangledown U\,]\!)_F \in F^{\star} \gtrsim F^{\star} \bullet F^{\star} \\ \Leftarrow & U \circ F.F^{\star}.I \in F^{\star} \gtrsim F \bullet F^{\star} \quad . \end{array}$$

Proof We construct $R \in F^{\star} \gtrsim F^{\star} \bullet F^{\star}$ using theorem 58 as follows:

$$\begin{array}{ll} & R \in F^{\star} \gtrsim F^{\star} \bullet F^{\star} \\ \Leftarrow & \qquad \{ \ \text{theorem 58}, \bullet \quad R \ = \ (\![\, S\,]\!)_F \ \} \\ & S \circ I + F.F^{\star}.I \in F^{\star} \gtrsim F^{\star} + F \bullet F^{\star} \\ \Leftarrow & \qquad \{ \ \bullet \quad S \ = \ T \triangledown U \ \} \\ & T \in F^{\star} \gtrsim F^{\star} \ \wedge \ U \circ F.F^{\star}.I \in F^{\star} \gtrsim F \bullet F^{\star} \\ \Leftarrow & \qquad \{ \ \bullet \quad T \ = \ F^{\star}.I \ \} \\ & U \circ F.F^{\star}.I \in F^{\star} \gtrsim F \bullet F^{\star} \quad . \end{array}$$

□

Theorem 60 Let τ_F and η_F be defined by

$$\begin{array}{lll}
\text{(a)} & \tau_F \ = \ \hookrightarrow\,, & \text{and} \\
\text{(b)} & \eta_F \ = \ \hookleftarrow \circ F.F^{\star}.I & .
\end{array}$$

Then $(F^{\star},\ \tau_F,\ \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F)$ is a monad.

Proof Since $F^{\star} \ = \ \mathbb{I} + F \bullet F^{\star}$ it is clear that $\tau_F \in F^{\star} \gtrsim \mathbb{I}$. Also, since $\eta_F \in F^{\star} \gtrsim F \bullet\ F^{\star}$, it follows from lemma 59 that $\llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \in F^{\star} \gtrsim F^{\star} \bullet F^{\star}$. It remains to verify the unit and associativity laws. For the associative law we have

$$\begin{array}{ll}
 & \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ F^{\star}.\llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
= & \qquad \{ \ \text{map fusion: theorem 213} \ \} \\
 & \llparenthesis\, F^{\star}.I \triangledown \eta_F \ \circ \ \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F + F.I \,\rrparenthesis_F \\
= & \qquad \{ \ \text{junc-sum fusion: 119,} \\
 & \qquad\quad \text{domain of } \eta_F\text{, domain of } \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \} \\
 & \llparenthesis\, \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \triangledown \ \eta_F \,\rrparenthesis_F \qquad .
\end{array}$$

Hence,

$$\begin{array}{ll}
 & \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ F^{\star}.\llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
 & = \quad \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
\Leftarrow & \qquad \{ \ \text{catamorphism fusion: 170} \ \} \\
 & \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ F^{\star}.I \triangledown \eta_F \\
 & = \quad \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \triangledown \eta_F \ \circ \ I + F.\llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
\equiv & \qquad \{ \ \text{junc-sum fusion: 119, junc cancellation: 133(a)} \ \} \\
 & \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ F^{\star}.I \quad = \quad \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
 & \wedge \ \ \llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \ \circ \ \eta_F \quad = \quad \eta_F \ \circ \ F.\llparenthesis\, F^{\star}.I \triangledown \eta_F \,\rrparenthesis_F \\
\equiv & \qquad \{ \ \text{domain of catamorphisms: 162,} \\
 & \qquad\quad \text{definition of } \eta_F\text{, computation rules: 160(a) and 124(b)} \ \} \\
 & \textbf{true} \quad .
\end{array}$$

The verification of the unit law is equally straightforward and left to the reader.

□

Note that $\llparenthesis\, F^{\star}.I \triangledown \hookleftarrow \,\rrparenthesis_F$ could also be taken for the multiplier and that this choice would be more in keeping with the truly polymorphic spirit of the spec calculus.

The following theorem appears in the paper "Adjunctions in the spec calculus" published in the 1991 workshop proceedings and is reproduced here for the convenience of the reader.

Theorem 61 (Sum Adjunction)

(a) $(R \circ \hookrightarrow \in F \gtrsim G \;\wedge\; R \circ \hookleftarrow \in F \gtrsim H)$
$\Leftarrow \quad R \circ I + I \in F \gtrsim G + H \quad .$

(b) $R \in F \gtrsim G \wedge S \in F \gtrsim H \;\Rightarrow\; R \triangledown S \in F \gtrsim G + H \quad .$

Proof Straightforward expansion of the definition of $\gtrsim$ and use of the computation and domain rules for disjoint sum.

□

Acknowledgement

Many thanks to Harold Weffers for

- LaTeXing the bulk of this document, including the creation of the david star
- preparing the figures,
- assisting with the construction of the Haskell program.