Categorical Fixed Point Calculus

Roland Backhouse, Marcel Bijsterveld,
Rik van Geldrop and Jaap van der Woude*

May 1995

Abstract

A number of lattice-theoretic fixed point rules are generalised to category theory and applied to the construction of isomorphisms between list structures.

1 Introduction

Category theoreticians view a preordered set as a particular sort of category in which there is at most one arrow between any pair of objects. According to this view, several concepts of lattice theory are instances of concepts of category theory as shown in table 1.

An alternative viewpoint, advocated by Lambek [9], is that lattice theory is a valuable source of inspiration for novel results in category theory. Indeed, it is our view that for the purposes of advancing programming methodology category theory may profitably be regarded as "coherently constructive lattice theory[1]". That is to say, arrows between objects of a category may be seen as "witnesses" to a preordering between the objects. Category theory is thus "constructive" because it is a theory about how to construct such witnesses rather than a theory solely about their existence. Category theory is "coherently constructive" because it is also a theory about the relations between such witnesses (i.e. the existence of commuting diagrams and naturality properties). Adopting this view of category theory, the theory's contribution to programming methodology can be likened to the contribution of constructive type theory, viz. the emphasis on program construction as a by-product of the manipulation of types.

This paper is a contribution to the practical application of these ideas. In the paper we develop a number of "fixed-point rules" in category theory each of which is inspired by (and generalises) a fixed-point rule in lattice theory. We then apply these rules to the construction of a number of elementary but fundamental isomorphisms between list structures. A number of the rules we derive appear to be new but a more important

*Department of Computing Science, Eindhoven University of Technology, The Netherlands.

[1]It has been remarked that we should say "preorder" theory rather than "lattice" theory. From the point of view of computing science, however, a category without sums and products has little relevance. Thus, it is indeed lattice theory that is our source of inspiration.

Lattice theory concept	is an instance of the category theory concept
preorder	category
monotonic function	functor
(pointwise) ordering between functions	natural transformation between functors
supremum	colimit
least	initial
Galois connection	adjunction
prefix point	algebra
closure operator	monad

Table 1: Lattice theory versus category theory

contribution may be to have collected them together and illustrated their application in equational reasoning about list structures.

2 Notation and Terminology

In this section we give a brief summary of our notational conventions.

Suppose $\mathcal{C}$ is a category. Then, we write $x{\in}\mathcal{C}$ to denote that x is an object of the category $\mathcal{C}$, and $f \in x \overset{\mathcal{C}}{\longleftarrow} y$ when f is an arrow in $\mathcal{C}$ with codomain x and domain y. The identity arrow on object x is denoted by id_x . The identity endofunctor on category $\mathcal{C}$ is denoted by $\mathsf{Id}_{\mathcal{C}}$, or just Id if it is clear which category is intended. Typically, the composition of arrows f and g is denoted by $f{\circ}g$ (irrespective of the category), whereby

$$f \in x{\leftarrow}y \wedge g \in y{\leftarrow}z \;\Rightarrow\; f{\circ}g \in x{\leftarrow}z \quad .$$

For functors F and G , however, we denote their composition by $F{\bullet}G$. Application of functor F to (object or arrow) x is denoted with an infix dot: thus, $F.x$. The category of functors to category $\mathcal{C}$ from category $\mathcal{D}$ is denoted $\mathsf{Fun}(\mathcal{C},\mathcal{D})$. Given a functor $F \in \mathcal{C}{\leftarrow}\mathcal{D}$ and a category $\mathcal{E}$, we can construct two functors $(F{\bullet}) \in \mathsf{Fun}(\mathcal{C},\mathcal{E}){\leftarrow}\mathsf{Fun}(\mathcal{D},\mathcal{E})$ and $({\bullet}F) \in \mathsf{Fun}(\mathcal{E},\mathcal{D}){\leftarrow}\mathsf{Fun}(\mathcal{E},\mathcal{C})$. The definition of ($F{\bullet}$) is as follows. For every functor $G{\in}\mathsf{Fun}(\mathcal{D},\mathcal{E})$ we have:

$$(F{\bullet}).G = F{\bullet}G \quad .$$

Application of ($F{\bullet}$) to the natural transformation $\eta \in G{\leftarrow}H$, where $G{,}H \in \mathsf{Fun}(\mathcal{D},\mathcal{E})$, is denoted by $F{\bullet}\eta$ and defined pointwise by

$$(F{\bullet}\eta)_x = F.\eta_x \quad \text{for each object } x \text{ in } \mathcal{E}.$$

Similarly, for every functor $G{\in}\mathsf{Fun}(\mathcal{E},\mathcal{C})$ we have:

$$({\bullet}F).G = G{\bullet}F \quad .$$

Application of $(\,\bullet F\,)$ to the natural transformation $\eta \in G \leftarrow H$, where $G,H \in \mathsf{Fun}(\mathcal{E},\mathcal{C})$, is denoted by $\eta \bullet F$ and defined pointwise by

$$(\eta \bullet F)_x = \eta_{F.x} \quad \text{for each object } x \text{ in } \mathcal{D}.$$

The standard notation is $F\eta$ and ηF. One advantage of explicitly denoting the composition is that we can use multiple-character identifiers to name natural transformations and functors. This we do often, with the general convention that sans serif identifiers are constants denoting (specific) natural transformations.

Application of functors always takes precedence over composition of arrows unless parentheses dictate otherwise. For example, $F \bullet \eta \circ \tau \bullet G$ denotes $(F \bullet \eta) \circ (\tau \bullet G)$ according to this convention, where the pointwise composition of natural transformations is denoted by the usual arrow composition. We assume familiarity with the algebraic properties of such expressions. Where such properties are used we refer to them with the hint "Godement's rules".

We shall have occasion to refer to the arrow category $\mathsf{Arr}.\mathcal{C}$ of $\mathcal{C}$. The objects of $\mathsf{Arr}.\mathcal{C}$ are the arrows in $\mathcal{C}$ and the arrows of $\mathsf{Arr}.\mathcal{C}$ are commuting squares in $\mathcal{C}$, i.e.

$$(\varphi, \psi) \in f \overset{\mathsf{Arr}.\mathcal{C}}{\longleftarrow} g \;\equiv\; f \circ \psi = \varphi \circ g \quad .$$

Note that, for functors F and G to $\mathcal{C}$ from $\mathcal{D}$, a natural transformation $\eta \in F \leftarrow G$ is a functor to $\mathsf{Arr}.\mathcal{C}$ from $\mathcal{D}$, defined on arrows by $\eta.f = (F.f\,,\,G.f)$. In fact these two notions are equivalent: any functor to $\mathsf{Arr}.\mathcal{C}$ from $\mathcal{D}$ is a natural transformation between its components. This also justifies the dot notation $\eta \bullet H$, introduced above, as a composition of functors.

Following Malcolm [14, 15] and Fokkinga [5] we denote the unique arrow in a category $\mathcal{C}$ to an object x from an initial object a by the "banana bracket notation" $(\!|\mathcal{C}\,;\, x =\!:\, a|\!)$. That is, a is initial in $\mathcal{C}$ if and only if, for all arrows f and all objects x in $\mathcal{C}$,

$$f \in x \overset{\mathcal{C}}{\longleftarrow} a \;\equiv\; f = (\!|\mathcal{C}\,;\, x =\!:\, a|\!) \quad .$$

The notion in category theory that corresponds to the notion of a prefix point in lattice theory is known as an *algebra*[2]. Suppose $\mathcal{C}$ is a category and F is an endofunctor on $\mathcal{C}$. An *F-algebra* is an arrow in $\mathcal{C}$ of type $x \leftarrow F.x$ for some object x of $\mathcal{C}$. The codomain of an F-algebra is referred to as the *carrier* of the algebra. The category $\mathcal{C}$ will be called the *base category*.

The F-algebras are organised into a category $\mathsf{Alg}.F$ as follows. The objects are the F-algebras and the arrows φ to F-algebra f from F-algebra g are characterised by the equation:

$$\varphi \in f \overset{\mathsf{Alg}.F}{\longleftarrow} g \;\equiv\; \varphi \in \mathsf{cod}.f \overset{\mathcal{C}}{\longleftarrow} \mathsf{cod}.g \;\wedge\; f \circ F.\varphi = \varphi \circ g \quad .$$

(Here cod denotes the codomain functor to the base category from $\mathsf{Alg}.F$.) Alternatively, the category $\mathsf{Alg}.F$ may be defined as the subcategory of $\mathsf{Arr}.\mathcal{C}$ consisting of arrows of the

[2]The definition we are about to give is weaker than that given in [12] and [11], the terminological confusion that this leads to having been deplored by Lambek [10]. Nevertheless it seems to have become standard among computing scientists. See, for example, [13].

form $(\varphi\,,\,F.\varphi)$. The codomain functor on $\mathsf{Alg}.F$ is then the (suitably restricted) "first component" functor to $\mathcal{C}$ from $\mathsf{Arr}.\mathcal{C}$.

An *initial F-algebra* is an initial object in the category $\mathsf{Alg}.F$. Existence of initial F-algebras is harder to predict than the existence of least prefix points. Generalisations to category theory of the Knaster-Tarski theorem have been considered by Lambek [9, 10] (and others). For our purposes it will suffice to assume that all the initial algebras we require do indeed exist. In particular, we will often assume that a given endofunctor, F , has a canonical initial algebra, denoted by $\mathsf{mu}F$. The carrier (i.e. codomain) of this canonical algebra will be denoted by μF . So, $\mathsf{mu}F \in \mu F \leftarrow F.\mu F$. This choice of notation facilitates comparison of our categorical fixed point theorems with the lattice theoretic theorems that they generalise.

Using the banana-bracket notation, the unique arrow to F-algebra α from initial F-algebra β is denoted $(\!|\mathsf{Alg}.F\,;\,\alpha =:\ \beta|\!)$. We usually abbreviate this to $(\!|F\,;\,\alpha =:\ \beta|\!)$. Sometimes, when there is absolutely no question of what is intended, we omit the first argument and write $(\!|\alpha\ =:\ \beta|\!)$.

Adjunctions are sometimes defined in terms of the unit and counit [12] of the adjunction and sometimes in terms of a natural isomorphism between hom-sets [7]. In the case of the latter definition it is useful to have a name for the two components of the isomorphism: we use the terms *lower* and *upper adjungate* and we denote them using the floor and ceiling operators as suggested by Fokkinga [5] . Thus, if $F\in\mathcal{C}\leftarrow\mathcal{D}$ and $G\in\mathcal{D}\leftarrow\mathcal{C}$ are adjoint functors, F being the lower adjoint, and $f \in x \overset{\mathcal{C}}{\longleftarrow} F.y$, then $\lceil f\rceil_{x,y} \in G.x \overset{\mathcal{D}}{\longleftarrow} y$. Similarly, if $g \in G.x \overset{\mathcal{D}}{\longleftarrow} y$ then $\lfloor g\rfloor_{x,y} \in x \overset{\mathcal{C}}{\longleftarrow} F.y$.

Finally, the inverse of an isomorphism f (between two objects in a category) will be denoted by $f^{\cup}$.

3 Fixed Point Calculus

In this section we present four basic fixed point theorems, and several theorems that are each the result of combining two or more of the basic theorems. The basic fixed point theorems that we present are respectively: *the fusion rule*, *the rolling rule*, *the abstraction theorem* and *the diagonal rule*.

The fusion rule combines the concept of an initial algebra with the concept of an adjunction. The rolling rule generalises the property (commonly known to category theoreticians as "Lambek's lemma" [9]) that initial F-algebras are fixed points of F . (Its namesake in lattice theory generalises the property that a least prefix point of monotonic function f is a fixed point of f .) The rolling rule is too elementary to be called "important" in its own right but it is extremely useful in combination with the other rules. An instance is the *exchange rule* which is a combination of the rolling rule with the fusion rule. The exchange rule is so called because it states when two lower adjoints may be exchanged in the construction of initial algebras. A second instance of the rolling rule is the *iterated square theorem*.

The abstraction theorem, combines the concept of an initial algebra with the concept of parameterisation. The last basic fixed point rule, the diagonal rule, captures the basic principle of decomposing the construction of an initial algebra into the construction of a succession of such algebras.

In lattice theory, the fusion rule appears in a slightly different form in the work of Cousot and Cousot [3], the abstraction theorem and exchange rules seem to be novel, and the rolling and diagonal rules seem to be "folklore" (i.e. we do not know to whom they should be credited, but they are widely known). Most are straightforward exercises to anyone versed in lattice theory but we know of no publication in which all are stated, let alone (a subset of) their applications presented.

In category theory, the fusion rule has been derived independently by Hermida and Jacobs [8]; the abstraction theorem and exchange rules seem to be novel. The rolling rule (in the form given here) has been derived by Lambert Meertens in unpublished discussion notes. The diagonal rule is stated and proved for ω-categories in [13] but we are not aware of any publication stating the theorem at the same level of generality as here.

In lattice theory, the abstraction and fusion theorem can be easily combined to prove a theorem dubbed "beautiful" by Dijkstra and Scholten [4, p. 159]. In category theory, the same combination of the abstraction and fusion theorems leads to a similarly "beautiful" theorem, an important special instance of which is that ω-cocontinuity is preserved by the process of constructing initial algebras [16, p.289]. The derivation of this theorem is used as an illustration of our view of category theory as coherently constructive lattice theory in [2].

Categorical fixed point rules have previously been studied by Freyd [6]. Freyd defines a category to be *algebraically complete* if all endofunctors on the category have initial algebras. He then proves that the product of two algebraically complete categories is algebraically complete and observes a rolling rule as a corollary. An intermediate result is the iterated square theorem mentioned earlier. These two theorems, included here for purposes of comparison, are corollaries of our basic theorems. Indeed, because Freyd makes the blanket assumption of algebraic completeness, we are able to establish stronger versions of Freyd's theorems. In particular our rolling and diagonal rules are stronger than what can be derived from Freyd's theorems. See section 3.5 for further discussion.

For space reasons we omit proofs of all the rules. Complete proofs are given in [1].

3.1 The Fusion Theorem

The fusion theorem is in fact an immediate corollary of the following theorem:

Theorem 1 Let $(F \in \mathcal{C} \leftarrow \mathcal{D}\,,\, F^\sharp \in \mathcal{D} \leftarrow \mathcal{C})$ be an adjunction and let $G \in \mathcal{D} \leftarrow \mathcal{D}$ and $H \in \mathcal{C} \leftarrow \mathcal{C}$ be functors . Assume also that $\mathsf{swap} \in F \bullet G \cong H \bullet F$ is a natural isomorphism. Then, there is an adjunction between the categories $\mathsf{Alg}.H$ and $\mathsf{Alg}.G$.

Specifics Let unit and counit denote the unit and co-unit, and $\lfloor\ \rfloor$ and $\lceil\ \rceil$ denote the lower and upper adjungates, of the adjunction $(F \in \mathcal{C} \leftarrow \mathcal{D}\,,\, F^\sharp \in \mathcal{D} \leftarrow \mathcal{C})$, respectively. Then the isomorphism swap gives rise to an isomorphism $\mathsf{adjswap} \in F^\sharp \bullet H \cong G \bullet F^\sharp$ defined by

$$\mathsf{adjswap}_x = \lfloor (H \bullet \mathsf{counit} \circ \mathsf{swap}^\cup \bullet F^\sharp)_x \rfloor \quad .$$

The functor $K \in \mathsf{Alg}.H \leftarrow \mathsf{Alg}.G$ defined by

$$K.g = F.g \circ \mathsf{swap}_{\mathsf{cod}.g} \quad \text{, where } g \in \mathsf{Alg}.G,$$

$$K.\varphi = F.\varphi \quad \text{, where } \varphi \text{ is an arrow in } \mathsf{Alg}.G$$

is a lower adjoint and the upper adjoint is the functor $K^\sharp \in \mathsf{Alg}.G \leftarrow \mathsf{Alg}.H$ defined by

$$K^\sharp.h = F^\sharp.h \circ \mathsf{adjswap}_{\mathsf{cod}.h} \quad \text{, where } h \in \mathsf{Alg}.H$$

$$K^\sharp.\psi = F^\sharp.\psi \quad \text{, where } \psi \text{ is an arrow in } \mathsf{Alg}.H$$

The left adjungate of this adjunction is defined on arrows ψ in $\mathsf{Alg}.H$ by $\lfloor \psi \rfloor$. The right adjungate is defined similarly.
□

Theorem 2 (Fusion Rule) With the same assumptions as in theorem 1 and the additional assumption that $\mathsf{Alg}.G$ has an initial object $\mathsf{mu}G$, we have that

$$F.\mathsf{mu}G \circ \mathsf{swap}_{\mu G}$$

is an initial object in the category $\mathsf{Alg}.H$. So, for every initial H-algebra, $\mathsf{mu}H$,

$$F.\mathsf{mu}G \circ \mathsf{swap}_{\mu G} \cong \mathsf{mu}H \quad .$$

As a consequence we also have an isomorphism in the base category, i.e.

$$F.\mu G \cong \mu H \quad .$$

Specifics In both isomorphisms

$$(\!|F.\mathsf{mu}G \circ \mathsf{swap}_{\mu G} \quad =: \; \mathsf{mu}H|\!)$$

is the arrow to $F.\mu G$ from μH and

$$\lceil (\!|F^\sharp.\mathsf{mu}H \circ \mathsf{adjswap}_{\mu H} \quad =: \; \mathsf{mu}G|\!) \rceil_{\mathsf{mu}H,\mathsf{mu}G}$$

is its inverse.
□

Although we don't give proofs of the fixed point rules the details given in the "specifics" section of each theorem can be seen as a trace of the proofs: Each isomorphism is constructed by a "mutual containment" argument whereby the arrows witness the individual containments and each component of the arrows witnesses a step in the proof. Occurrences of composition, for example, witness the use of transitivity of the ordering relation and occurrences of the banana brackets witness the minimality of the given algebra. The proof of the fusion theorem consists thus of a constructive proof of the inclusions $F.\mu G \sqsupseteq \mu H$ and $\mu H \sqsupseteq F.\mu G$ followed by a verification of the fact that the witnesses are inverses of each other.

In subsequent applications of the fusion theorem we will not need to know the details of the witnesses to the isomorphism, all that we need to know being that the isomorphisms exist. Let us therefore abbreviate $(\!|F.\mathsf{mu}G \circ \mathsf{swap}_{\mu G} \;\; =: \; \mathsf{mu}H|\!)$ to $\mathsf{fuse}_{F,G,H}.\mathsf{swap}$. The rule we use in future applications is thus (assuming the conditions of the fusion theorem are satisfied):

(3) $\mathsf{fuse}_{F,G,H}.\mathsf{swap} \in F.\mu G \cong \mu H$.

3.2 Rolling, Square and Exchange Rules

Theorem 4 (Rolling Rule) Let $F \in \mathcal{C} \leftarrow \mathcal{D}$ and $G \in \mathcal{D} \leftarrow \mathcal{C}$ be functors. Suppose that $\mathsf{mu}(G \bullet F)$ is an initial $(G \bullet F)$-algebra. Then $F.\mathsf{mu}(G \bullet F)$ is an initial $(F \bullet G)$-algebra. Thus, for every initial $(F \bullet G)$-algebra, $\mathsf{mu}(F \bullet G)$,

$$F.\mathsf{mu}(G \bullet F) \cong \mathsf{mu}(F \bullet G) \quad .$$

As a consequence we also have an isomorphism in the base category, i.e.

$$F.\mu(G \bullet F) \cong \mu(F \bullet G) \quad .$$

Specifics In both isomorphisms

$$(\!|F.\mathsf{mu}(G \bullet F) \;\; =: \; \mathsf{mu}(F \bullet G)|\!)$$

is the arrow to $F.\mu(G \bullet F)$ from $\mu(F \bullet G)$ and

$$\mathsf{mu}(F \bullet G) \circ F.(\!|G.\mathsf{mu}(F \bullet G) \;\; =: \; \mathsf{mu}(G \bullet F)|\!)$$

is its inverse.
□

Letting $\mathsf{roll}_{F,G}$ denote $(\!|F.\mathsf{mu}(G \bullet F) \;\; =: \; \mathsf{mu}(F \bullet G)|\!)$ we thus have:

(5) $\mathsf{roll}_{F,G} \in F.\mu(G \bullet F) \cong \mu(F \bullet G)$.

Substituting F for G we find that $\mathsf{mu}(F^2) \circ F.(\!|F^2\,;\; F.\mathsf{mu}(F^2) =: \mathsf{mu}(F^2)|\!)$ is an F-algebra (where F^2 denotes $F \bullet F$). Freyd [6] observes that this is an initial F-algebra, thus establishing the existence of an initial F-algebra given the existence of an initial F^2-algebra. He calls this theorem the *iterated square theorem*. He also observes that the existence of an initial F-algebra guarantees the existence of an initial F^2-algebra provided that the category has products. The corresponding theorems in lattice theory are that a prefix point of (monotonic endofunction) F is a prefix point of F^2, and that $x \sqcap F.x$ is a prefix point of F whenever x is a prefix point of F^2 (in a preorder having infima). In particular, $\mu(F^2) \cong \mu F$. The precise statement of Freyd's theorem is as follows.

Theorem 6 (Iterated square) Let F be an endofunctor of $\mathcal{C}$ such that $\mathsf{mu}(F^2)$ exists. Then $\mathsf{mu}(F^2)\circ F.(\!|F^2\,;\ F.\mathsf{mu}(F^2) =:\ \mathsf{mu}(F^2)|\!)$ is an initial F-algebra, the unique arrow from which to F-algebra f being $(\!|\mathsf{Alg}.F^2\,;\ f\circ F.f =:\ \mathsf{mu}(F^2)|\!)$. Moreover, if the category $\mathcal{C}$ has products, an initial F-algebra, $\mathsf{mu}F$, induces an initial F^2-algebra, namely $\mathsf{mu}F\circ F.\mathsf{mu}F$.

□

The final theorem in this section combines the rolling rule with the fusion theorem.

Theorem 7 (Exchange Rule) Given are three functors $F\in\mathcal{C}\leftarrow\mathcal{D}$, $G\in\mathcal{D}\leftarrow\mathcal{C}$ and $H\in\mathcal{D}\leftarrow\mathcal{C}$ such that G and H are lower adjoints in adjunctions. Furthermore, we have the isomorphism $\mathsf{mirror}\in H{\bullet}F{\bullet}G\cong G{\bullet}F{\bullet}H$. Finally, we assume that an initial $F{\bullet}G$ algebra, $\mathsf{mu}(F{\bullet}G)$, and an initial $F{\bullet}H$ algebra, $\mathsf{mu}(F{\bullet}H)$, exist. Then

$$\mu(F{\bullet}G)\cong\mu(F{\bullet}H)\quad .$$

Specifics In this isomorphism

$$(\!|\mathsf{roll}_{F,G}\circ F.\mathsf{fuse}_{H,(F\bullet G),(G\bullet F)}.\mathsf{mirror}\quad =:\ \mathsf{mu}(F{\bullet}H)|\!)$$

is the witness to $\mu(F{\bullet}G)$ from $\mu(F{\bullet}H)$ and

$$(\!|\mathsf{roll}_{F,H}\circ F.\mathsf{fuse}_{G,(F\bullet H),(H\bullet F)}.(\mathsf{mirror}{\cup})\quad =:\ \mathsf{mu}(F{\bullet}G)|\!)$$

is its inverse.

□

Note that the exchange rule does not give an isomorphism between algebras, only between the carriers.

Denoting the witness to the isomorphism to $\mu(F{\bullet}G)$ from $\mu(F{\bullet}H)$ by $\mathsf{exch}_{F,G,H}.\mathsf{mirror}$, we have the rule that

$$(8)\qquad \mathsf{exch}_{F,G,H}.\mathsf{mirror}\in\mu(F{\bullet}G)\cong\mu(F{\bullet}H)\quad ,$$

provided of course that F, G, H and mirror satisfy the conditions in the exchange rule.

There is a second exchange rule in which the order of F and G, and F and H, is reversed. A third rule, involving four functors instead of three, combines both. The exchange rule given here is the one that we use in section 4.

3.3 The Abstraction Theorem

A standard way of constructing functors inductively uses abstraction from an argument of a binary functor. Let $\oplus\in\mathcal{D}\leftarrow\mathcal{C}\times\mathcal{D}$ be a binary functor. Then $\oplus$ induces a binary functor (for an arbitrary category $\mathcal{E}$)

$$\dot{\oplus}\in\mathsf{Fun}(\mathcal{D},\mathcal{E})\leftarrow\mathsf{Fun}(\mathcal{C},\mathcal{E})\times\mathsf{Fun}(\mathcal{D},\mathcal{E})$$

by defining it as the composition of $\oplus\bullet$ with the embedding $\bullet\triangle_{\mathcal{E}}$ of $\mathsf{Fun}(\mathcal{C},\mathcal{E})\times\mathsf{Fun}(\mathcal{D},\mathcal{E})$ in $\mathsf{Fun}(\mathcal{C}\times\mathcal{D},\mathcal{E})$, where $\triangle_{\mathcal{E}} \in \mathcal{E}\times\mathcal{E} \leftarrow \mathcal{E}$ is the diagonal functor. Thus

$$F\dot{\oplus}G = \oplus \bullet(F,G)\bullet \triangle_{\mathcal{E}} \quad \text{and} \quad \eta\dot{\oplus}\nu = \oplus \bullet(\eta,\nu)\bullet \triangle_{\mathcal{E}} \quad .$$

In particular, for an object x and object or arrow ξ ,

(9) $(F\dot{\oplus}G).\xi = F.\xi \oplus G.\xi \quad \text{and} \quad (\eta\dot{\oplus}\nu)_x = \eta_x \oplus \nu_x \quad .$

Since every (left) section $x\oplus$ of $\oplus$, defined by $(x\oplus).d = x\oplus d$ and $(x\oplus).f = \mathsf{id}_x\oplus f$, is an endofunctor on $\mathcal{D}$, we may consider their algebras. Assuming the existence of initial $x\oplus$-algebras, for all x, and fixing them to

$$\mathsf{mu}(x\oplus) \in \mu(x\oplus) \leftarrow x\oplus\mu(x\oplus)$$

we can construct the *initial algebra functor* $\mathsf{mu}_\oplus \in \mathsf{Arr}.\mathcal{D} \leftarrow \mathcal{C}$ by defining

(10) $f^* = (\!|\mathsf{mu}(y\oplus)\circ f\oplus\mathsf{id}_{\mu(x\oplus)} \ =:\ \mathsf{mu}(x\oplus)|\!)$,

for $f \in y \overset{\mathcal{C}}{\longleftarrow} x$,

(11) $\mathsf{mu}_\oplus.x = \mathsf{mu}(x\oplus) \quad \text{and} \quad \mathsf{mu}_\oplus.f = (f^*\,,\, f\oplus f^*) \quad .$

The corresponding carrier functor, $\mathsf{cod}\bullet\mathsf{mu}_\oplus$, (i.e. the codomain functor after the initial algebra functor) is the well known *map* functor which we denote by $\varpi_\oplus$. That is, for all objects x and y and all arrows $f \in x\leftarrow y$,

(12) $\varpi_\oplus.x = \mu(x\oplus) \quad \text{and} \quad \varpi_\oplus.f = f^* \quad .$

In the case of (cons) lists, for example, we define the list functor as $\varpi_\oplus$ where the functor $\oplus$ maps the pair (x,y) to $\mathbb{1}+y\times x$.

The abstraction theorem states that a map functor is itself an initial algebra. More precisely, since $\mathsf{mu}_\oplus \in \mathsf{Arr}.\mathcal{D}\leftarrow\mathcal{C}$ and $\mathsf{mu}_\oplus.f = (\varpi_\oplus.f\,,\,(\mathsf{Id}\dot{\oplus}\varpi_\oplus).f)$ it follows from our earlier discussion of the correspondence between functors to arrow categories and natural transformations that the initial algebra functor is a natural transformation: $\mathsf{mu}_\oplus \in \varpi_\oplus \leftarrow \mathsf{Id}\dot{\oplus}\varpi_\oplus$, and thus it is an $(\mathsf{Id}\dot{\oplus})$-algebra. The theorem states that $\mathsf{mu}_\oplus$ is an initial $(\mathsf{Id}\dot{\oplus})$-algebra. In words: abstracting from the parameter in an initial algebra yields an initial algebra.

Theorem 13 (Abstraction Theorem) Let $\oplus \in \mathcal{D}\leftarrow\mathcal{C}\times\mathcal{D}$ be a binary functor written as infix operator. Assume also the existence of a canonical initial algebra $\mathsf{mu}(x\oplus)$ with codomain $\mu(x\oplus)$ for each object x in $\mathcal{C}$. Define the initial algebra functor $\mathsf{mu}_\oplus$ as in (12). Then $\mathsf{mu}_\oplus$ is an initial algebra for $\mathsf{Id}\dot{\oplus}$ (for arbitrary $\mathcal{E}$); moreover, if $\mathsf{mu}(\mathsf{Id}\dot{\oplus})$ denotes a particular initial $\mathsf{Id}\dot{\oplus}$-algebra having codomain $\mu(\mathsf{Id}\dot{\oplus})$, then

$$\mathsf{mu}(\mathsf{Id}\dot{\oplus}) \cong \mathsf{mu}_\oplus \quad \text{and} \quad \mu(\mathsf{Id}\dot{\oplus}) \cong \varpi_\oplus \quad .$$

Specifics In both isomorphisms, between $\mathsf{mu}(\mathsf{Id}\dot{\oplus})$ and $\mathsf{mu}_\oplus$ and their carriers,

$$(\!|\mathsf{mu}_\oplus \ =:\ \mathsf{mu}(\mathsf{Id}\dot{\oplus})|\!)$$

is the arrow to $\varpi_\oplus$ from $\mu(\mathsf{Id}\dot{\oplus})$ and the natural transformation α defined by

$$\alpha_z = (\!|(\mathsf{mu}(\mathsf{Id}\dot{\oplus}))_z \;\; =: \; \mathsf{mu}(z\oplus)|\!) \quad \text{for each object } z \text{ in } \mathcal{C}$$

is its inverse.
□
Note that the abstraction theorem can be generalised by replacing the functor Id by some functor $F \in \mathcal{C} \leftarrow \mathcal{E}$ and by defining the binary infix functor $\otimes$ as $x \otimes y = F.x \oplus y$. With the corresponding assumptions on $\otimes$ we have

(14) $\mathsf{mu}(\mathsf{Id}\dot{\otimes}) = \mathsf{mu}(F\dot{\oplus}) \cong \mathsf{mu}_\otimes$ and $\varpi_\oplus \bullet F = \mu(F\dot{\oplus}) \cong \varpi_\otimes$.

The witness is denoted by $\mathsf{abs}_{(F,\oplus)}$ so that our calculation rule is:

(15) $\mathsf{abs}_{(F,\oplus)} \in \varpi_\oplus \bullet F \cong \mu(F\dot{\oplus})$.

3.4 The Diagonal Rule

Theorem 16 (Diagonal Rule) Let $\oplus \in \mathcal{C} \leftarrow \mathcal{C} \times \mathcal{C}$ be a binary functor. Assume that for each object x in $\mathcal{C}$ an initial object, $\mathsf{mu}(x\oplus)$, exists in $\mathsf{Alg}.(x\oplus)$. Define functor ϖ[3] as in (10) and (12) and denote the unary functor $\mathsf{Id}\dot{\oplus}\mathsf{Id}$ by $\hat{\oplus}$. Then, if $\mathsf{mu}\hat{\oplus}$ is an initial object in $\mathsf{Alg}.\hat{\oplus}$,

$$(\!|\mathsf{mu}\hat{\oplus} \;\; =: \; \mathsf{mu}((\mu\hat{\oplus})\oplus)|\!)$$

is an initial object in the category $\mathsf{Alg}.\varpi$. So, for every initial ϖ-algebra, $\mathsf{mu}\varpi$,

$$(\!|\mathsf{mu}\hat{\oplus} \;\; =: \; \mathsf{mu}((\mu\hat{\oplus})\oplus)|\!) \cong \mathsf{mu}\varpi \quad .$$

Conversely, if $\mathsf{mu}\varpi$ is an initial object in $\mathsf{Alg}.\varpi$ then

$$\mathsf{mu}\varpi \circ \mathsf{mu}((\mu\varpi)\oplus) \circ \mathsf{id}_{\mu\varpi} \oplus (\mathsf{mu}\varpi)^{\cup}$$

is an initial object in the category $\mathsf{Alg}.\hat{\oplus}$. So, for every initial $\hat{\oplus}$-algebra, $\mathsf{mu}\hat{\oplus}$,

$$\mathsf{mu}\varpi \circ \mathsf{mu}((\mu\varpi)\oplus) \circ \mathsf{id}_{\mu\varpi} \oplus (\mathsf{mu}\varpi)^{\cup} \cong \mathsf{mu}\hat{\oplus} \quad .$$

As a consequence we also have an isomorphism in the base category: $\mu\hat{\oplus} \cong \mu\varpi$, i.e.

$$\mu(x \mapsto x \oplus x) \cong \mu(x \mapsto \mu(y \mapsto x \oplus y)) \quad .$$

Specifics In all three isomorphisms

$$(\!|(\!|\mathsf{mu}\hat{\oplus} \;\; =: \; \mathsf{mu}((\mu\hat{\oplus})\oplus)|\!) \;\; =: \; \mathsf{mu}\varpi|\!)$$

is the arrow to $\mu\hat{\oplus}$ from $\mu\varpi$ and

$$\mathsf{mu}\varpi \circ (\!|\mathsf{mu}((\mu\varpi)\oplus) \circ \mathsf{mu}\varpi \oplus \mathsf{id}_{\varpi.\mu\varpi} \;\; =: \; \mathsf{mu}\hat{\oplus}|\!)$$

[3]For convenience we omit the subscript on ϖ.

is its inverse.
□

As with the fusion and rolling rules we do not need the specific details of the witnessing isomorphisms. Introducing the abbreviation $\mathsf{diag}_{\oplus}$ for the isomorphism to $\mu\varpi$ from $\mu\hat{\oplus}$, the rule we apply in subsequent calculations is thus:

(17) $\mathsf{diag}_{\oplus} \in \mu(x\mapsto\mu(y\mapsto x\oplus y)) \cong \mu(x\mapsto x\oplus x)$.

3.5 Mutual Recursion

In the previous sections we have presented the four basic rules of the categorical fixed point calculus: the fusion rule, the rolling rule, the abstraction theorem and the diagonal rule. The topic of this section is the relationship between our basic rules, in particular the rolling rule and the diagonal rule, with Freyd's discussion of algebraically complete categories [6].

Before commencing the discussion let us first emphasise that the diagonal rule has *three* parts: (a) the existence of an initial $\hat{\oplus}$-algebra given an initial ϖ-algebra, (b) the existence of an initial ϖ-algebra given an initial $\hat{\oplus}$-algebra and (c) the isomorphism in the base category of the carriers of the two types of initial algebra.

Now, adapting Freyd's terminology, a preorder is *algebraically complete* if every monotonic endofunction on the preorder has a least prefix point. A well-known theorem (attributed by Winskel [19] to Bekic but without source reference) is that the product of two algebraically complete preorders is algebraically complete. Specifically, suppose $\mathcal{C}$ and $\mathcal{D}$ are algebraically complete preorders and $F \in \mathcal{C}\times\mathcal{D} \leftarrow \mathcal{C}\times\mathcal{D}$. Decompose F into two binary functions $\odot \in \mathcal{C} \leftarrow \mathcal{D}\times\mathcal{C}$ and $\otimes \in \mathcal{D}\leftarrow\mathcal{C}\times\mathcal{D}$ in such a way that $F.(x,y) = (y\odot x\,,\, x\otimes y)$. Then a least prefix point of F is $(\mu(x\mapsto p.x\odot x)\ ,\ \mu(y\mapsto q.y\otimes y))$ where $p.x = \mu(x\otimes)$ and $q.y = \mu(y\odot)$. We have shown in [17] that this *mutual recursion* theorem is a corollary of the diagonal rule and (lattice-theoretic) iterated square theorem. The same proof can be transformed to a proof of Freyd's theorem that the product of two algebraically complete categories is algebraically complete. See [1] for full details.

Rather than deriving the mutual recursion theorem from the diagonal and rolling rules, Freyd first establishes the mutual recursion theorem and then observes the rolling rule as a special case. His rolling rule is however weaker since he demands the existence of both an initial $(F\bullet G)$-algebra and an initial $(G\bullet F)$-algebra. Our rule only assumes the existence of one in order to guarantee the existence of the other.

This way of deriving a rolling role appears also in texts on programming language semantics. (See for example [19].) It is also quite easy to see that *part (c)* of the diagonal rule (the isomorphism between the carriers of the two types of initial algebra) is a consequence of the mutual recursion theorem (define $y\odot x$ to be y and $x\otimes y$ to be $x\oplus y$). This does not mean, however, that the diagonal rule is in any way weaker than the mutual recursion theorem. On the contrary, our rule is sharper in that it gives sharper conditions on the existence of initial algebras in a product category than Freyd's blanket assumption of the algebraic completeness of the component categories —using parts (a) and (b) of the diagonal rule— and we are not aware of any way of deriving those parts from Freyd's mutual recursion theorem.

In summary, the rolling and diagonal rules given here give tighter conditions on the existence of initial algebras and have Freyd's mutual recursion theorem as corollary.

4 Applications

In lattice theory the four basic fixed point rules amount to a very effective *equational* [4] fixed point calculus. (For a variety of examples see [17].) The categorical fixed point rules amount to a very effective equational *and constructive* fixed point calculus. That is to say, isomorphisms between type structures can be obtained as a by-product of equational arguments in the lattice-theoretic fixed point calculus. In this section we illustrate the method by deriving a number of isomorphisms between list structures. In each case a calculation in lattice theory is augmented with "witnesses" in a mechanical way.

In the calculations natural transformations (and isomorphisms) of n-ary functors play a dominant role and we need some notational conventions to deal with constant arguments. To that end we consider the product of categories (responsible for n-arity) to be associative and we use the connector $\vartriangle$ between functors to construct functors to an n-ary codomain. For example, . for $H_i \in \mathcal{D}_i \leftarrow \mathcal{C}$ the functor $H = H_0 \vartriangle H_1 \vartriangle H_2 \in \mathcal{D}_0 \times \mathcal{D}_1 \times \mathcal{D}_2 \leftarrow \mathcal{C}$ is defined by $H.x = (H_0.x, H_1.x, H_2.x)$ for objects as well as arrows. The usual product of (three) functors is denoted by (F, G, H).

Let η be a natural transformation between, say, ternary functors F and G and let H be a functor to their ternary domain, then $\eta \bullet H$ is a natural transformation between $F \bullet H$ and $G \bullet H$. A suitable H may change the arity. For example, the functor $H = \mathsf{K}.a \vartriangle \mathsf{K}.b \vartriangle \mathsf{Id}$, where $\mathsf{K}.a$ denotes the constant a functor, fixes the arguments a and b thus turning a unary functor into a ternary functor; similarly, $H' = \mathsf{K}.a \vartriangle (\mathsf{Id}, \mathsf{Id})$ fixes only a . Instead of $\eta \bullet H$ and $\eta \bullet H'$ we prefer the more suggestive notations $\eta_{a,b,\_}$ and $\eta_{a,\_,\_}$ respectively.

An example in the next section is a natural isomorphism leapF between binary functors, say $\ominus$ and $\odot$. Then $\mathsf{leapF}\bullet$ is a natural transformation between $\dot{\ominus}$ and $\dot{\odot}$ and, by the convention above,

$$(\mathsf{leapF}\bullet)_{\mathsf{Id},\_} = (\mathsf{leapF}\bullet) \bullet (\mathsf{K.Id} \vartriangle \mathsf{id}) \quad ,$$

$$((\mathsf{leapF}\bullet) \bullet (\mathsf{K.Id} \vartriangle \mathsf{id}))_G = \mathsf{leapF} \bullet (\mathsf{Id} \vartriangle G) \quad \text{and}$$

$$(\mathsf{leapF} \bullet (\mathsf{Id} \vartriangle G))_a = \mathsf{leapF}_{a,G.a} \quad .$$

4.1 Leapfrog Rules

There are several ways that lists can be defined: "cons" lists where elements are added ("consed" in Lisp terminology) to the beginning of a list , "snoc" lists where elements are

[4] Involving equalities only as opposed to proofs of equality via mutual containment.

added ("snocked" , the reverse of "cons") to the end of a list , and "join" lists where two lists are "joined" together. We consider only cons and snoc lists.

The cons list functor is the map functor $\varpi_{\oplus}$ (see (12) and (10)) where

$$x{\oplus}y = \mathbb{1}{+}(x{\times}y) \quad ,$$

and the snoc list functor is the map functor $\varpi_{\ominus}$ where

$$x{\ominus}y = \mathbb{1}{+}(y{\times}x) \quad .$$

In order to be able to consider both at once we abstract from their definitions in this section and consider the situation in which we are given a (unary) functor F and a binary functor $\otimes$, (replacing $\mathbb{1}{+}$ and $\times$, respectively), such that $x\otimes$ and $\otimes x$ are lower adjoints for every object x. We then define two map functors by

$$\hat{F} = \varpi_{F\bullet\oplus} \quad \text{and} \quad \check{F} = \varpi_{F\bullet\oplus\bullet\bowtie} \quad .$$

where the binary to binary functor $\bowtie$ interchanges the coordinates.

The cons list functor is an instantiation of both $\hat{F}$ and $\check{F}$ by choosing $F = \mathbb{1}{+}$ and $\otimes = \times$ and $\otimes = \times \bullet \bowtie$ respectively. Similarly, the snoc list functor is an instantiation of $\hat{F}$ and $\check{F}$ for the same F and interchanged choices for $\otimes$.

Our first application of the fixpoint rules states that $\hat{F}$ and $\check{F}$ are isomorphic functors provided that F obeys a so-called "leapfrog" property with respect to $\otimes$. When reading the proofs we recommend that the "witnesses" are ignored the first time around. (That is, ignore everything marked by a bullet and all membership information.) Stripped of this information the proof obtained is the lattice-theoretic proof.

Theorem 18 Suppose $\mathsf{Id}\dot{\otimes}$ and $\dot{\otimes}\mathsf{Id}$ are lower adjoints and we have the following natural isomorphism (i.e. natural in the parameters a and b)

$$\mathsf{leapF}_{a,b} \in F.(a{\otimes}b) \otimes a \cong a \otimes F.(b{\otimes}a) \quad .$$

Define the map functor $\hat{F}$ to be $\varpi_{\oplus}$ and the map functor $\check{F}$ to be $\varpi_{\ominus}$ where

$$\oplus = F\bullet\otimes \quad \text{and} \quad \ominus = F\bullet\otimes\bullet\bowtie \quad .$$

Then $\hat{F}\cong\check{F}$.

(It is useful to have a catchy name for important properties. We call the existence of the isomorphism leapF in the above theorem a *leapfrog* property because the parameter a "leapfrogs" from one side to the other of the functor F .)

Proof

$$
\begin{array}{ll}
 & \alpha \in \hat{F}\cong\check{F} \\
\Leftarrow & \qquad \{ \qquad \text{abstraction;} \bullet \quad \alpha \;=\; \mathsf{abs}_{\mathsf{Id},\oplus}\circ\beta\circ(\mathsf{abs}_{\mathsf{Id},\ominus})^{\cup} \quad \} \\
 & \beta \in \mu(\mathsf{Id}\dot{\oplus})\cong\mu(\mathsf{Id}\dot{\ominus}) \\
\equiv & \qquad \{ \qquad \mathsf{Id}\dot{\oplus} = (F\bullet)\bullet\mathsf{Id}\dot{\otimes} \;\text{ and }\; \mathsf{Id}\dot{\ominus} = (F\bullet)\bullet\dot{\otimes}\mathsf{Id} \quad \}
\end{array}
$$

$$\beta \in \mu((F\bullet)\bullet(\mathsf{Id}\dot{\otimes})) \cong \mu((F\bullet)\bullet(\dot{\otimes}\mathsf{Id}))$$

$\Leftarrow$ { exchange rule, $\mathsf{Id}\dot{\otimes}$ and $\dot{\otimes}\mathsf{Id}$ are lower adjoints

• $\beta = \mathsf{exch}_{F\bullet,\mathsf{Id}\dot{\otimes},\dot{\otimes}\mathsf{Id}}.\gamma$ }

$$\gamma \in (\dot{\otimes}\mathsf{Id})\bullet(F\bullet)\bullet(\mathsf{Id}\dot{\otimes}) \cong (\mathsf{Id}\dot{\otimes})\bullet(F\bullet)\bullet(\dot{\otimes}\mathsf{Id})$$

$\Leftarrow$ { $((\dot{\otimes}\mathsf{Id})\bullet(F\bullet)\bullet(\mathsf{Id}\dot{\otimes})).G = (F\bullet(\mathsf{Id}\dot{\otimes}G))\dot{\otimes}\mathsf{Id}$

$((\mathsf{Id}\dot{\otimes})\bullet(F\bullet)\bullet(\dot{\otimes}\mathsf{Id})).G = \mathsf{Id}\dot{\otimes}(F\bullet(G\dot{\otimes}\mathsf{Id}))$

(See the discussion preceding this subsection.) }

$$\gamma = (\mathsf{leapF}\bullet)_{\mathsf{Id},\_} \quad .$$

Thus,

$$\mathsf{abs}_{\mathsf{Id},\oplus} \circ \mathsf{exch}_{F\bullet,\mathsf{Id}\dot{\otimes},\dot{\otimes}\mathsf{Id}}.(\mathsf{leapF}\bullet)_{\mathsf{Id},\_} \circ (\mathsf{abs}_{\mathsf{Id},\ominus})^{\cup} \in \hat{F} \cong \check{F} \quad .$$

□

The construction of $\hat{F}$ from F can be repeatedly applied giving $\hat{\hat{F}}$, etc. Our next application shows that this process preserves the leapfrog property provided that $\otimes$ is also associative (up to isomorphism).

Theorem 19 (Leapfrog Preservation) Suppose F is a functor and $\otimes$ is a binary functor such that $a\otimes$ and $\otimes a$ are lower adjoints for every object a. Define the map functor $\hat{F}$ to be $\varpi_{F\bullet\otimes}$. Suppose we have two natural isomorphisms

$$\mathsf{ass}_{a,b,c} \in (a\otimes b)\otimes c \cong a\otimes(b\otimes c)$$

and

$$\mathsf{leapF}_{a,b} \in F.(a\otimes b)\otimes a \cong a\otimes F.(b\otimes a) \quad .$$

Then

$$\hat{F}.(a\otimes b)\otimes a \cong a\otimes \hat{F}.(b\otimes a) \quad .$$

Proof Letting $G = F\bullet(a\otimes b)\otimes$, $H = F\bullet(b\otimes a)\otimes$ and $L = a\otimes\bullet F\bullet b\otimes$ for brevity in the subscripts of fuse we have:

$$\alpha \in \hat{F}.(a\otimes b)\otimes a \cong a\otimes \hat{F}.(b\otimes a)$$

$\equiv$ { definition of $\hat{F}$ }

$$\alpha \in (\otimes a).\mu(F\bullet(a\otimes b)\otimes) \cong (a\otimes).\mu(F\bullet(b\otimes a)\otimes)$$

$\Leftarrow$ { invent intermediate $\mu((a\otimes)\bullet F\bullet(b\otimes))$; • $\alpha = \beta\circ\gamma^{\cup}$ }

$$\beta \in (\otimes a).\mu(F\bullet((a\otimes b)\otimes)) \cong \mu((a\otimes)\bullet F\bullet(b\otimes))$$

$$\wedge\ \gamma \in (a\otimes).\mu(F\bullet((b\otimes a)\otimes)) \cong \mu((a\otimes)\bullet F\bullet(b\otimes))$$

$$\Leftarrow \quad \{ \quad \text{fusion;} \bullet \quad \beta = \mathsf{fuse}_{\otimes a,G,L}.\varphi \text{ and } \gamma = \mathsf{fuse}_{a\otimes,H,L}.\psi \quad \}$$

$$\varphi \in \otimes a \bullet F \bullet (a\otimes b)\otimes \cong a\otimes \bullet F \bullet b\otimes \bullet \otimes a$$

$$\wedge \ \psi \in a\otimes \bullet F \bullet (b\otimes a)\otimes \cong a\otimes \bullet F \bullet b\otimes \bullet a\otimes$$

$$\Leftarrow \quad \{ \quad \mathsf{ass}_{p,q,\_} \in (p\otimes q)\otimes \cong p\otimes \bullet q\otimes \text{ and}$$

$$\mathsf{ass}_{p,\_,q} \in \otimes q \bullet p\otimes \cong p\otimes \bullet \otimes q$$

$$\bullet \quad \varphi = (\otimes a \bullet F \bullet \mathsf{ass}_{a,b,\_}) \circ \xi \circ (a\otimes \bullet F \bullet \mathsf{ass}_{b,\_,a})$$

$$\bullet \quad \psi = a\otimes \bullet (F \bullet \mathsf{ass}_{b,a,\_}) \quad \}$$

$$\xi \in \otimes a \bullet F \bullet a\otimes \bullet b\otimes \cong a\otimes \bullet F \bullet \otimes a \bullet b\otimes$$

$$\Leftarrow \quad \{ \quad \text{definition } \mathsf{leapF} \quad \}$$

$$\xi = (\mathsf{leapF}_{a,\_}) \bullet (b\otimes)$$

We conclude that

$$\mathsf{fuse}_{\otimes a,G,L}.((\otimes a \bullet F \bullet \mathsf{ass}_{a,b,\_}) \circ ((\mathsf{leapF}_{a,\_}) \bullet (b\otimes)) \circ (a\otimes \bullet F \bullet \mathsf{ass}_{b,\_,a}))$$
$$\circ \quad (\mathsf{fuse}_{a\otimes,H,L}.(a\otimes \bullet (F \bullet \mathsf{ass}_{b,a,\_})))^{\cup} \quad .$$

witnesses the isomorphism of $\hat{F}.(a\otimes b) \otimes a$ and $a \otimes \hat{F}.(b\otimes a)$.
□
Note that in order to combine theorems 18 and 19 to show that (for example) $\hat{\hat{F}}$ and $\check{\check{F}}$ are isomorphic we need to show that the isomorphism constructed in the latter theorem is natural in the parameters a and b. For general F this is a likely to be an impossible task but in section 4.3 we argue why this is immediately the case for the "Kleene" functors.

4.2 Lists

In this section, we prove isomorphisms between certain list structures and simultaneously construct the witnesses. The first two are simple instantiations of the leapfrog rules presented in the last subsection.

Formally, we assume that the base category is a bicartesian, exponential category [7]. The specific details of this assumption are as follows. First, denoting the product of a and b by $a{\times}b$ and their sum by $a{+}b$ (both of which are assumed to exist), we assume for all objects y the functors $y\times$ and $\times y$, i.e. the functor $\times$ with the left or right argument fixed to y, have upper adjoints. Second, denoting the terminal object of the base category by $\mathbb{1}$, we assume the existence of the following isomorphisms. For all objects a, b and c,

$$\mathsf{lunit}_a \in \mathbb{1}{\times}a \cong a \quad ,$$

$$\mathsf{runit}_a \in a{\times}\mathbb{1} \cong a \quad ,$$

$$\mathsf{sumass}_{a,b,c} \in (a{+}b){+}c \cong a{+}(b{+}c) \quad ,$$

$$\mathsf{proass}_{a,b,c} \in (a{\times}b){\times}c \cong a{\times}(b{\times}c) \quad ,$$

$$\mathsf{rdist}_{a,b,c} \in (a{+}b){\times}c \cong (a{\times}c){+}(b{\times}c) \quad ,$$

$$\mathsf{ldist}_{a,b,c} \in a{\times}(b{+}c) \cong (a{\times}b){+}(a{\times}c) \quad .$$

In order to instantiate the fusion theorem it suffices to know that the lower adjungate of the adjunction with $\times y$ as lower adjoint (and exponentiation as upper adjoint) is the operation known as "currying" to functional programmers, the upper adjungate is "uncurrying", and the counit is function evaluation.

Example 20 Defining the Conslist functor and the Snoclist functor by

$$\mathsf{Clist} = a \mapsto \mu(y \mapsto \mathbb{1} + a{\times}y), \quad \mathsf{Slist} = a \mapsto \mu(y \mapsto \mathbb{1} + y{\times}a)$$

we have the functor isomorphism

$$\mathsf{Clist} \cong \mathsf{Slist} \quad .$$

Proof Instantiate in theorem 18:

$$F \;:=\; \mathbb{1}{+} \quad , \quad \otimes \;:=\; \times \quad , \mathsf{ass} \;:=\; \mathsf{proass}$$

and

$$\mathsf{leapF}_{a,b} \;:=\; \mathsf{rdist}_{\mathbb{1},a\times b,a} \circ ((\mathsf{lunit}_a \circ \mathsf{runit}_a^{\cup}){+}\mathsf{proass}_{a,b,a}) \circ \mathsf{ldist}_{a,\mathbb{1},b\times a}^{\cup} \quad .$$

□

From now on we consider cons lists only; the cons list functor will be denoted by an asterisk. That is, we assume the functor $*$ is defined to be the map functor $\varpi_{\oplus}$ (see (12) and (10)) where

$$x{\oplus}y = \mathbb{1}{+}(x{\times}y) \quad .$$

In contrast to normal functor applications we will omit the dot when the functor $*$ is applied to an object.

Example 21

$$a{\times} * (b{\times}a) \cong\; * (a{\times}b){\times}a \quad .$$

Proof Instantiate in theorem 19:

$$F \;:=\; \mathbb{1}{+} \quad , \quad \otimes \;:=\; \times \quad , \mathsf{ass} \;:=\; \mathsf{proass}$$

and

$$\mathsf{leapF}_{a,b} \;:=\; \mathsf{rdist}_{\mathbb{1},a\times b,a} \circ ((\mathsf{lunit}_a \circ \mathsf{runit}_a^{\cup}){+}\mathsf{proass}_{a,b,a}) \circ \mathsf{ldist}_{a,\mathbb{1},b\times a}^{\cup} \quad .$$

□

Note that we can apply the leapfrog theorem once again with F instantiated to $*$ (provided naturality is proven; see the last section). In language theory nothing new is obtained — because $\hat{*}$ and $*$ are equal. In category theory we do obtain a new theorem — because $\hat{*}$ and $*$ are not even isomorphic. Thus the leapfrog theorem has an infinite number of applications! (The equality between $\hat{*}$ and $*$ in language theory boils down to the equality between x and $x{+}x$ which isn't constructively valid.)

The final example has been chosen for its relative difficulty, and its practical relevance. We call it the *list decomposition problem*. An instance is the so-called "lines-unlines" problem: given a sequence of two types of characters, delimiters and non-delimiters, write a program to divide the sequence into (possibly empty) "lines" of non-delimiters seperated by single delimiters. Construct in addition the inverse of the program.

In the following calculation we employ a notation whereby the witnesses to isomorphisms are included in the hints marked by a bullet ("$\bullet$"). Specifically a proof step of the form

$$
\begin{array}{ll}
 & F \\
\cong & \{ \quad \text{hint}, \bullet \quad \alpha \quad \} \\
 & G
\end{array}
$$

is short for

$$\alpha \in F \cong G \Leftarrow \text{hint} \quad .$$

The list composition problem, expressed as an isomorphism between datatypes, boils down to showing that the star decomposition theorem of regular languages is constructively valid.

Example 22 (List Decomposition)

$$*a \times *(b \times *a) \cong *(b{+}a) \quad .$$

Proof

$$
\begin{array}{ll}
 & *a \times *(b \times *a) \\
= & \{ \quad \text{definition of } * \quad \} \\
 & *a \times \mu(y \mapsto \mathbb{1} + (b \times *a) \times y) \\
\cong & \{ \quad \text{Godement's rules}, \bullet \quad \mathsf{id}_{*a} \times \mathsf{fuse}.((\mathbb{1}+)\bullet(\mathsf{proass}_{b,*a,\_})) \quad \} \\
 & *a \times \mu(y \mapsto \mathbb{1} + b \times (*a \times y)) \\
\cong & \{ \quad \text{rolling rule}, \bullet \quad \mathsf{roll}_{*a\times,F} \\
 & \quad\ \ \text{where } F \text{ denotes the functor } y \mapsto \mathbb{1} + b \times y \quad \} \\
 & \mu(y \mapsto *a \times (\mathbb{1} + b \times y)) \\
\cong & \{ \quad \text{fusion rule}, \bullet \quad \mathsf{fuse}.((\mathsf{listfuse}_{a,\_})\bullet F) \quad \} \\
 & \mu(y \mapsto \mu(z \mapsto (\mathbb{1} + b \times y) + a \times z))
\end{array}
$$

$$
\begin{array}{ll}
\cong & \{ \quad \text{diagonal rule,} \bullet \quad \mathsf{diag}_{\oplus} \text{ , where } y{\oplus}z = (1\!\!1 + b{\times}y) + a{\times}z \quad \} \\
& \mu(y \mapsto (1\!\!1 + b{\times}y) + a{\times}y) \\
\cong & \{ \quad \text{fusion rule,} \bullet \quad \mathsf{fuse}.((\mathsf{sumass}_{1\!\!1,\_,\_}){\bullet}(b{\times}\ ,\ a{\times}) \circ (1\!\!1{+}){\bullet}(\mathsf{rdist}^{\cup}_{b,a,\_})) \quad \} \\
& \mu(y \mapsto 1\!\!1 + (b{+}a){\times}y) \\
= & \{ \quad \text{definition of } * \quad \} \\
& *(b{+}a) \quad .
\end{array}
$$

We conclude that,

$$
\begin{array}{ll}
& \mathsf{id}_{*a} \times \mathsf{fuse}.((1\!\!1{+}){\bullet}(\mathsf{proass}_{b,*a,\_})) \\
\circ & \mathsf{roll}_{*a\times,F} \\
\circ & \mathsf{fuse}.((\mathsf{listfuse}_{a,\_}){\bullet}F) \\
\circ & \mathsf{diag}_{\oplus} \\
\circ & \mathsf{fuse}.((\mathsf{sumass}_{1\!\!1,\_,\_}){\bullet}(b{\times}\ ,\ a{\times}) \circ ((1\!\!1{+}){\bullet}(\mathsf{rdist}^{\cup}_{b,a,\_}))) \\
\in & *a{\times}*(b{\times}*a) \cong *(b{+}a) \quad .
\end{array}
$$

where $F = y \mapsto 1\!\!1 + b{\times}y$ and $y{\oplus}z = (1\!\!1 + b{\times}y) + a{\times}z$.
□

4.3 "Theorems for Free"

In the list decomposition example we have proved an isomorphism in the base category for all instantiations of the objects a and b. Formally, however, we have not shown that we have an isomorphism between the binary functors whose object parts are

$$a{,}b \mapsto *a{\times}*(b{\times}*a) \quad ,$$

and

$$a{,}b \mapsto *(b{+}a) \quad .$$

We have yet to prove that the constructed isomorphisms are natural in the parameters a and b. The same remark can be made about the list leapfrog example. Yet it is well known that naturality is what Wadler [18] has dubbed a "theorem for free". In this section we briefly explain why this theorem is "for free" and, specifically, the role of the abstraction theorem in that claim.

The key is to note that all statements about a bicartesian exponential category can be lifted to statements about a functor category by pointwise parameterisation [12, theorem 1, p.111]. Specifically, assume that a bicartesian, exponential category $\mathcal{C}$ is given, with coproduct and product operators $+$ and $\times$. As discussed in the preamble to the abstraction theorem, these operators can be lifted to $\dot{+}$ and $\dot{\times}$ for arbitrary domain categories $\mathcal{E}$; i.e. $\dot{+}$ and $\dot{\times}$ are binary operators on the category Fun . They are, in fact, the coproduct and the product in Fun: by the parameterised limit theorem it follows that $(F\dot{+}G, \mathsf{inl}_{F,G}, \mathsf{inr}_{F,G})$ is the coproduct of F and G, where $(\mathsf{inl}_{F,G})_x = \mathsf{inl}_{F.x,G.x}$ and similarly for $\mathsf{inr}_{F,G}$. In the same way $\dot{\times}$ is the binary product functor on Fun and also the adjoints of $F\dot{\times}$ and $\dot{\times}F$ can be defined. The category Fun is thus a bicartesian, exponential category where, for instance,

$$\mathsf{sumass}_{F,G,H} \in (F\dot{+}G)\dot{+}H \cong F\dot{+}(G\dot{+}H)$$

is given by

$$\mathsf{sumass}_{F,G,H} = \mathsf{sumass} \bullet (F \vartriangle G \vartriangle H) \quad .$$

That is,

$$(\mathsf{sumass}_{F,G,H})_x = \mathsf{sumass}_{F.x,G.x,H.x} \quad .$$

The abstraction theorem admits a similar result for $*$ and for map functors in general.

Suppose F is a functor. Define functor $\oplus$ by $a \oplus y = \mathbb{1} + F.a \times y$. Then we observe that

$$\begin{array}{cl}
 & *\bullet F \\
= & \{ \quad \text{definition } * \text{ and composition} \quad \} \\
 & a \mapsto \mu(a\oplus) \\
= & \{ \quad \text{abstraction theorem} \quad \} \\
 & \mu(G \mapsto \mathsf{Id}\dot{\oplus}G) \\
= & \{ \quad (\mathsf{Id}\dot{\oplus}G).x = x \oplus G.x = \mathbb{1}+(F.x \times G.x) \\
 & \qquad = (\mathsf{K}.\mathbb{1} \dot{+} F\dot{\times}G).x \text{, extensionality} \quad \} \\
 & \mu(G \mapsto \mathsf{K}.\mathbb{1} \dot{+} F\dot{\times}G) \quad .
\end{array}$$

If we now define the functor $\dot{*}$ on objects F by $\mu(G \mapsto \mathsf{K}.\mathbb{1} \dot{+} F\dot{\times}G)$ we can reformulate this observation as

(23) $\quad * \bullet F = \dot{*}F \quad .$

The list decomposition theorem now becomes a theorem in the functor category whereby each functor is replaced by its "dotted" version. That is, the list decomposition theorem constructs an isomorphism $\mathsf{decomp}_{F,G}$ satisfying

$$\mathsf{decomp}_{F,G} \in \dot{*}F\dot{\times}\dot{*}(G\dot{\times}\dot{*}F) \cong \dot{*}(F\dot{+}G) \quad .$$

Moreover, we can now use abstraction to obtain the required isomorphism between the two *functors* rather than a collection of isomorphisms between objects.

$$
\begin{array}{ll}
 & a,b \mapsto \ast\, a \times \ast\, (b \times \ast\, a) \\
= & \{ \quad \text{Introducing the functors } Exl \text{ and } Exr\text{, where } Exl.(a,b) = a \text{ and } \\
 & \quad\quad Exr.(a,b) = b \quad \} \\
 & a,b \mapsto \ast\, Exl.(a,b) \times \ast\, (Exr.(a,b) \times \ast\, Exl.(a,b)) \\
= & \{ \quad \text{abstraction} \quad \} \\
 & (\ast \bullet Exl) \dot{\times} (\ast \bullet (Exr \dot{\times} (\ast \bullet Exl))) \\
= & \{ \quad \text{abstraction: (23)} \quad \} \\
 & \dot{\ast} Exl \dot{\times} \dot{\ast} (Exr \dot{\times} \dot{\ast} Exl) \\
\cong & \{ \quad \text{theorem 22} \bullet \quad \mathsf{decomp}_{Exl,Exr} \quad \} \\
 & \dot{\ast} (Exr \dot{+} Exl) \\
= & \{ \quad \text{abstraction} \quad \} \\
 & a,b \mapsto \ast\, (b + a) \quad .
\end{array}
$$

If full details of the definition of $\mathsf{decomp}_{Exl,Exr}$ are required then we would have to instantiate the witnesses in the statement of (22) in the following way:

$$a,b \ := \ Exl,Exr \quad ,$$

$$+, \times, \ast \ := \ \dot{+}, \dot{\times}, \dot{\ast}$$

and

$$\mathbb{1} \ := \ \mathsf{K}.\mathbb{1} \quad .$$

Simplification would then yield the witness obtained earlier.

Acknowledgement

We are grateful to the referees for pointing out Freyd's work to us.

References

[1] R. C. Backhouse, M. Bijsterveld, R. van Geldrop, and J.C.S.P. van der Woude. Category theory as coherently constructive lattice theory. Department of Mathematics and Computing Science, Eindhoven University of Technology. Working document. Available via world-wide web at `http://www.win.tue.nl/win/cs/wp/papers`, 1995.

[2] R.C. Backhouse and M. Bijsterveld. Category theory as coherently constructive lattice theory: an illustration. Technical report, Department of Computing Science, Eindhoven University of Technology, 1994. Available via world-wide web at `http://www.win.tue.nl/win/cs/wp/papers`.

[3] Patrick Cousot and Radhia Cousot. Systematic design of program analysis frameworks. In *Conference Record of the Sixth Annual ACM Symposium on Principles of Programming Languages*, pages 269–282, San Antonio, Texas, January 1979.

[4] E.W. Dijkstra and C.S. Scholten. *Predicate Calculus and Program Semantics*. Springer-Verlag, Berlin, 1990.

[5] Maarten M. Fokkinga. Calculate categorically! *Formal Aspects of Computing*, 4:673–692, 1992.

[6] Peter Freyd. Algebraically complete categories. In G. Rosolini A. Carboni, M.C. Pedicchio, editor, *Category Theory, Proceedings, Como 1990*, volume 1488 of *Lecture Notes in Mathematics*, pages 95–104. Springer-Verlag, 1990.

[7] P.J. Freyd and A. Scedrov. *Categories, Allegories*. North-Holland, 1990.

[8] Claudio A. Hermida and Bart Jacobs. An algebraic view of structural induction. Extended Abstract, May 1994.

[9] J. Lambek. A fixpoint theorem for complete categories. *Mathematische Zeitschrift*, 103:151–161, 1968.

[10] J. Lambek. Least fixpoints of endofunctors of cartesian closed categories. *Mathematical Structures in Computer Science*, 3:229–257, 1993.

[11] J. Lambek and P.J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[12] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1971.

[13] D.J. Lehman and M.B. Smyth. Algebraic specification of data types: A synthetic approach. *Math. Syst. Theory*, 14(2):97–140, 1981.

[14] G. Malcolm. *Algebraic data types and program transformation*. PhD thesis, Groningen University, 1990.

[15] G. Malcolm. Data structures and program transformation. *Science of Computer Programming*, 14(2–3):255–280, October 1990.

[16] E.G. Manes and M.A. Arbib. *Algebraic Approaches to Program Semantics*. Texts and Monographs in Computer Science. Springer-Verlag, Berlin, 1986.

[17] Eindhoven University of Technology Mathematics of Program Construction Group. Fixed point calculus. *Information Processing Letters*, 53(3):131–136, February 1995.

[18] P. Wadler. Theorems for free! In *4'th Symposium on Functional Programming Languages and Computer Architecture, ACM, London*, September 1989.

[19] G. Winskel. *???* ???, ???