A Class of Commuting Relators

Roland Backhouse, Henk Doornbos and Paul Hoogendijk
Department of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513,
5600 MB Eindhoven,
The Netherlands.

September 15, 1992

Abstract

The zip function is well known to functional programmers. Informally, zip maps two lists of the same length into a single list of pairs whereby $([a_1, b_1, ...], [a_2, b_2, ...]) \mapsto [(a_1, b_1), (a_2, b_2), ...]$. Zip is just one of a whole family of functions that *commute* the order of two data structures. Whilst zip commutes a pair of lists into a list of pairs, it is not difficult to imagine generalising zip to a function that commutes m lists each of length n into n lists each of length m. Indeed, this latter function is also well known under the name *matrix transposition*. With a little bit more imagination several other members of the family suggest themselves. For example, there is a function that commutes a tree of lists all of the same length into a list of trees all of the same shape. There is also a function that commutes a rose of trees all of the same shape into a tree of roses all of the same shape.

This paper considers the construction of such functions. We specify the class of so-called "zippable" data-structures and consider in detail a large and significant subclass of this class. For this subclass we show how to construct "zips" and identify several consequences of their being "zippable".

1 Introduction

The zip function is well known to functional programmers. Informally, zip maps two lists of the same length into a single list of pairs whereby $([a_1, b_1, ...], [a_2, b_2, ...]) \mapsto [(a_1, b_1), (a_2, b_2), ...]$.

Zip is just one of a whole family of functions that *commute* the order of two data structures. Whilst zip commutes a pair of lists into a list of pairs, it is not difficult to imagine generalising zip to a function that commutes m lists each of length n into n lists each of length m. Indeed, this latter function is also well known under the name *matrix transposition.* With a little bit more imagination several other members of the family suggest themselves. For example, there is a function that commutes a tree of lists all of the same length into a list of trees all of the same shape. There is also a function that commutes a rose of trees all of the same shape into a tree of roses all of the same shape. In this paper we explore the construction of such functions.

The construction of individual members of this class of functions (for example the zip function on pairs of lists or matrix transposition) is a straightforward programming exercise. Constructing, in one go, a *class* of such functions, having an infinite number of elements, is much more exacting, and demands a good theory of datatypes that facilitates the formulation of a compact and workable generic specification of the members of the class of functions. This paper illustrates the use of a *relational* theory of datatypes to achieve the task. We show how the class of functions can be constructed using the so-called *generator* of a datatype (a concept introduced in this paper), and how it can be constructed by purely syntactic calculation (and thus without reference to operational arguments).

2 The Requirement

The first problem we have to tackle is that of formulating a specification of the family of "zip" functions. The identification of data structures with the notion of relator is a major advance but is not quite enough. We will be obliged to introduce a couple of additional concepts.

Looking again at the examples above, the first step towards an abstract problem specification is clear enough. Replacing "list", "tree" etc. by "relator F" the problem is to construct a spec $zip_{F,G}$ for given relators F and

G, that maps F–structures of (G–structures of the same shape) into G–structures of (F–structures of the same shape). The last example cited above (roses of trees) fits this requirement directly. The first example — zipping a pair of lists — also fits if we associate cartesian product with pairing and if the definition of shape is such that all pairs have the same shape and two lists have the same shape if and only if they have the same length.

A second requirement, obtained by inspection of the examples above, which we may or may not choose to impose, is that $zip_{F,G}$ is a *bijection*. Certainly the zip function on lists is a bijection between lists of pairs and pairs of lists of the same length. However the transposition operator on a list of lists is only a bijection if restricted to *non–empty* lists. (The reason is that all lists, whatever their length, of empty lists are transposed to the empty list; thus transposition is not injective.) We shall indeed impose the requirement that $zip_{F,G}$ is a bijection and restrict ourselves to bottom–strict relators.

A third requirement is that $zip_{F,G}$ be truly *polymorphic*. That is to say, its application to a F-G–structure should not in any way depend on the values in that structure. We are thus saddled with the problem of defining the notion of "shape" for an arbitrary relator F and with saying precisely what we mean by "a polymorphic bijection mapping F–structures of (G–structures of the same shape) to G–structures of (F–structures of the same shape)".

3 Shapes and Generators

To ascertain whether two lists have the same length it is not necessary to compute their length. Instead, choose some arbitrary value, "$*$" say, and replace every element in the two lists by that value. Then compare the two lists of $*$'s for equality. Unlike the notion of length this process can easily be extended to arbitrary data structures. To see, for example, whether two trees have the same shape replace each element in the trees by "$*$" and then compare the two trees of $*$'s for equality.

In the spec calculus there is an even simpler way of expressing that two F–structures, for relator F, have the same shape. Recall that $F.R$ "maps" spec R over a given F–structure. (For example, $[\![\ List.R\ ]\!]$[1] is a relation

[1] $[\![R]\!]$ stands for the interpretation of the spec R in the relational model.

between lists such that for $a = [a_0, a_1, ...]$ and $b = [b_0, b_1, ...]$,

$$a \,[\![\, List.R \,]\!]\, b \;\equiv\; \#a = \#b \;\wedge\; \forall(i :: a_i \,[\![\, R]\!]\, b_i) \quad .$$

Since $[\![\top\!\top]\!]$ is the universal relation, $[\![\, F.\top\!\top \,]\!]$ is a relation that holds between two F-structures if and only if they have the same shape. (Compare this with $[\![\, F.I \,]\!]$ which is a relation holding between x and y if and only if x and y are equal F–structures.)

Some examples are $List.\top\!\top$, $\top\!\top + \top\!\top$ and $\top\!\top \times \top\!\top$. The interpretation of $List.\top\!\top$ is a relation holding between lists of equal length. The interpretation of $\top\!\top + \top\!\top$ is a relation consisting of all pairs of objects of the form (**inl**.x, **inl**.y) or (**inr**.x, **inr**.y) for some x and y. Thus elements of a disjoint sum have the same shape if they belong to the same component of the summation.

The interpretation of $\top\!\top \times \top\!\top$ is similarly the relation holding between all pairs "of the same shape". But pairs are "shapeless" — i.e. *all* pairs have the same shape! Formally,

$$(1) \qquad \top\!\top \times \top\!\top \;=\; I \times I \circ \top\!\top \circ I \times I$$

Note that the spec on the right of this equality is interpreted as the relation that holds between two objects if and only if they are both pairs. In comparison,

$$(2) \qquad \top\!\top + \top\!\top \;\neq\; I + I \circ \top\!\top \circ I + I$$

and

$$(3) \qquad List.\top\!\top \;\neq\; List.I \circ \top\!\top \circ List.I$$

In both (2) and (3) the spec on the right is greater than that on the left of the equation. This explains why in the definition of *zip* on pairs of lists there is an asymmetry in the description of its left and right domains.

We now know how to describe the shape of F–structures but how we do we capture the notion of "F–structures of (G–structures of the same shape)"? The mechanism we use involves the so–called *generator* associated with a relator.

We motivate the notion of generator by considering, as a first example, cartesian product. Consider the spec $I \vartriangle I$. The interpretation of $I \vartriangle I$ is a function that maps argument x into the pair (x, x). Thus $[\![I \vartriangle I]\!]$ generates a pair — an element of a product type — of which the two components are equal. Being equal is, of course, a very special case of having the same shape.

More generally, $R \vartriangle R$ is a spec whose interpretation is a relation that holds between (x, y) and z whenever $x[\![R]\!]z$ and $y[\![R]\!]z$. In particular, for arbitrary relator F, $F.\top\!\top \vartriangle F.\top\!\top$ is a spec whose interpretation is a relation that holds between (x, y) and z whenever all three are F–structures of the same shape. Most importantly, x and y are F–structures of the same shape. We say that $F.\top\!\top \vartriangle F.\top\!\top$ *generates* pairs of F–structures of the same shape.

To summarise, we call $I \vartriangle I$ the *primitive product generator* — it generates pairs in which the component elements are equal. The split operator is the *product generator* — it generates pairs. Note that $R \vartriangle S = R \times S \circ I \vartriangle I$. So the product generator (split) can always be expressed in terms of the composition of a product and the primitive product generator. The spec $R \vartriangle R$ generates pairs that have the same R–shape — i.e. pairs (x, y) for which some z exists with $x[\![R]\!]z$ and $y[\![R]\!]z$. Finally, $F.\top\!\top \vartriangle F.\top\!\top$ generates pairs of F–structures with the same shape — i.e. "product–structures of (F–structures of the same shape)".

Exercise 4 Split enjoys certain properties that are characteristic of generators. One has already been mentioned, the fusion property

(5) $\quad R \vartriangle S \;=\; R \times S \circ I \vartriangle I \;.$

The more general fusion property

(6) $\quad R \times S \circ T \vartriangle U \;=\; (R \circ T) \vartriangle (S \circ U)$

follows from (5) and is also characteristic of generators. Another property that we have not yet seen but is characteristic of a generator is

(7) $\quad \top\!\top \vartriangle \top\!\top \;=\; \top\!\top \times \top\!\top \circ \top\!\top.$

Prove this property.

A final property of split that is also characteristic of a generator is:

(8) $\quad I \vartriangle I \circ f \;=\; f \vartriangle f$

for all imps f.

Property (1) above is a property of split that is *not* characteristic of generators. The exercise is to prove (1). (Hint: note that
$\ll^{\cup} \circ \top\!\top \circ \ll \;=\; I \times I \circ \top\!\top \circ I \times I \;=\; \gg^{\cup} \circ \top\!\top \circ \gg$)
$\square$

Let us now consider lists. The *primitive list generator*, *Listgen*, would be a spec that, given a "seed" x, generates all lists of x's (the lists $[\,], [x], [x,x], ...$). Let us accept for the moment that such a spec can be defined within the calculus. (We show how to construct it very shortly.) Then the function $\hat{L}$ defined by

$$\hat{L}.R \quad = \quad \mathit{List}.R \circ \mathit{Listgen}$$

is called the *list generator*. To see what its interpretation is we note that, for list xs and element x

$$(9) \qquad xs[\![\mathit{Listgen}]\!]x \quad \equiv \quad \forall(i : 0 \leq i < \#xs : xs_i = x)$$

and thus, for all lists ys

$$
\begin{array}{ll}
 & ys[\![\hat{L}.R]\!]x \\
\equiv & \qquad \{ \text{ definition of } \hat{L} \text{ and interpretation of } \circ \ \} \\
 & \exists(xs :: ys[\![\mathit{List}.R]\!]xs \wedge xs[\![\ \mathit{Listgen}\ ]\!]x) \\
\equiv & \qquad \{ \text{ interpretation of } \mathit{List}.R \text{ and } (9) \ \} \\
 & \quad \forall(i :\ 0 \leq i < \#ys :\ ys_i[\![R]\!]xs_i) \\
 & \wedge \quad \forall(i :\ 0 \leq i < \#xs :\ xs_i = x) \\
\equiv & \qquad \{ \text{ calculus } \} \\
 & \forall(i :\ 0 \leq i < \#ys :\ ys_i[\![R]\!]x) \quad .
\end{array}
$$

i.e.

$$(10) \qquad ys[\![\hat{L}.R]\!]x \quad \equiv \quad \forall(i :\ 0 \leq i < \#ys :\ ys_i[\![R]\!]x) \quad .$$

Now let us substitute $\mathit{List}.\top\!\top$ for R. We get

$$
\begin{array}{ll}
 & yss[\![\hat{L}.\mathit{List}.\top\!\top]\!]xs \\
\equiv & \qquad \{ \ (10) \ \} \\
 & \forall(i :\ 0 \leq i < \#yss :\ yss_i[\![\mathit{List}.\top\!\top]\!]xs) \\
\equiv & \qquad \{ \text{ interpretation of } \mathit{List}.\top\!\top \ \} \\
 & \forall(i :\ 0 \leq i < \#yss :\ \#(yss_i) \ = \ \#xs)
\end{array}
$$

i.e. $\hat{L}.\mathit{List}.\top\!\top$ generates all lists of lists of equal length. Similarly $\hat{L}.(\top\!\top \times \top\!\top)$ generates all lists of pairs, and $\hat{L}.(\top\!\top + \top\!\top)$ generates all lists of one of the forms $[$**inl**$.a$, **inl**$.b$, ...$]$, $[$**inr**$.a$, **inr**$.b$, ...$]$, i.e. lists in which all elements are drawn from the same component of a disjoint sum type.

Generalising from this we conclude the following. Suppose F and G are relators. Suppose there is a spec $Fgen$ that given element x generates all F–structures built solely from x. Define

$$(11) \qquad \hat{F}.R \quad = \quad F.R \circ Fgen$$

for all specs R. Then $\hat{F}.G.\top\!\top$ is a spec whose interpretation is the generator of all F–structures of (G–structures of a given shape).
Another way of looking at it is as a non–deterministic function that given a G–structure x — the so-called *seed* — creates an F–structure with components all having the same shape as x.

4 The Specification

At this point we can return to the specification of $zip_{F,G}$ for given relators F and G. Our requirement can be summarised as the construction of a polymorphic bijection mapping F-structures of (G-structures of the same shape) to G-structures of (F-structures of the same shape). Since we are interested in the construction of a bijection we really want to construct two imps $zip_{F,G}$ and $zip_{G,F}$ and then show that $zip_{F,G} = (zip_{G,F})^{\cup}$. This observation allows us to relax the requirement on $zip_{F,G}$ somewhat. To be specific, we demand that $zip_{F,G}$ constructs G-structures of (F-structures of the same shape) from F-G-structures. (Note that totality is not required.) If $zip_{G,F}$ does the opposite — constructs F-structures of (G-structures of the same shape) from G-F-structures and $zip_{F,G} = (zip_{G,F})^{\cup}$ then it is clear that $zip_{F,G}$ must map F-structures of (G-structures of the same shape) to G-structures of (F-structures of the same shape).

Remark: The functional programmer's zip, mapping pairs of lists to lists of pairs, is so named to conjure up an analogy between its operational meaning and the operation of a zip on an article of clothing. (When a zip is pulled up corresponding teeth interlock; when the function zip is executed corresponding list elements are paired.) We have chosen to use the word "zip" to name our class of functions, but the metaphor now extends to the way in which zips are *manufactured.* Specifically, a zip (on an item of clothing) consists of two separate parts that, though not identical, are precisely interlocking. In what follows we construct our "zips" in two halves in such a way that the two halves, although not necessarily identical, are inverses of each other (i.e. "are precisely interlocking").

Of course, such metaphors may be quite misleading. So, if you find it amusing, all well and good. However, whatever your reaction to it please dismiss the metaphor from your mind when reading further!
End of Remark

Knowing that G-structures of (F-structures of the same shape) are generated from a given F-structure by $\hat{G}.F.\top\!\!\top$ and arbitrary F-G-structures are generated from a given F-structure by $F.\hat{G}.\top\!\!\top$ our first stab at a specification is

$$zip_{F,G} \circ F.\hat{G}.\top\!\!\top \quad = \quad \hat{G}.F.\top\!\!\top \quad .$$

This equation does indeed express the requirement that $zip_{F,G}$ construct G-structures of (F-structures of the same shape) from F-G-structures but it is not enough. It doesn't express the notion that *zip* be "polymorphic" — in particular, doesn't inspect or alter the values in the given F-G-structure. The problem is, however, easily resolved simply by replacing $\top\!\!\top$ in the above equation by an arbitrary spec R. The complete specification thus becomes:

Definition 12 (Zippable Relators) The ordered pair (F,G) of relators is said to be *zippable by* $zip_{F,G}$ if $zip_{F,G}$ is a spec with the following four properties:

$$(13) \qquad zip_{F,G} \circ F.\hat{G}.R \quad = \quad \hat{G}.F.R$$

$$(14) \qquad G.F.R \circ zip_{F,G} \quad = \quad zip_{F,G} \circ F.G.R \quad \text{for all specs } R \quad ,$$

$$(15) \qquad G.F.I \circ zip_{F,G} \quad = \quad zip_{F,G} \quad = \quad zip_{F,G} \circ F.G.I \quad .$$

$$(16) \qquad \mathrm{imp}\,.zip_{F,G} \quad .$$

□

Properties (13) and (14) specify that $zip_{F,G}$ maps F-G-structures into G-structures of (F-structures of the same shape) and is a natural transformation to $G \bullet F$ from $F \bullet G$[1]. (Being a natural transformation is what we mean with being "polymorphic".)

[1] $F \bullet G$ is defined by $(F \bullet G).R \quad = \quad F.(G.R)$.

Definition 17 (Commuting Relators) Relators F and G are said to be *commuting* if specs $zip_{F,G}$ and $zip_{G,F}$ exist such that the ordered pair (F,G) is zippable by $zip_{F,G}$, the ordered pair (G,F) is zippable by $zip_{G,F}$, and

(18) $\quad zip_{F,G} \;=\; (zip_{G,F})\cup \quad .$

□

Note that, if F and G are commuting relators, then the witness $zip_{F,G}$ is a bijection constructing all G-structures of (F-structures of the same shape) from all F-structures of (G-structures of the same shape).

Of course this is still not a specification because $\hat{F}$ has not been specified. For that three clauses are needed:

Definition 19 (Generator) The function $\hat{F}$ from specs to specs is called the *generator* of relator F if it satisfies the following three properties:

(20) $\quad \hat{F}.R \;=\; F.R \circ \hat{F}.I \quad$ for all spec R ,

(21) $\quad \hat{F}.\top\!\top \;=\; F.\top\!\top \circ \top\!\top \quad ,$

(22) $\quad \hat{F}.f \;=\; \hat{F}.I \circ f \quad$ for all imps f .

The *co-generator* of F, denoted $\check{F}$, is defined by

(23) $\quad \check{F} \;=\; (\hat{F}.(R\cup))\cup \quad .$

□

In (20) $\hat{F}.I$ is what we previously called the *primitive F-generator*. Note that from this equation it follows immediately that generators are monotonic functions. Property (21) expresses the idea that $\hat{F}.\top\!\top$ generates all F-structures independently of the given seed value. Property (22) says that if a function f is used in conjunction with the F-generator to generate F-structures from seed x, for some x, then this is the same as applying the primitive F-generator to seed $f.x$. (It can be shown that (20)–(22) has at least one solution and that in an *extensional* spec algebra it uniquely defines $\hat{F}$ but that is beyond our scope.)

Properties (13) and (15) of $zip_{F,G}$ and some domain calculus imply the following domain property

$$(24) \qquad (zip_{F,G})< \;=\; (\hat{G}.F.\top\!\!\top)< \quad .$$

This follows from the following calculation

$$\begin{array}{ll}
 & zip_{F,G} \circ \top\!\!\top \\
= & \qquad \{ \;(15)\; \} \\
 & zip_{F,G} \circ F.G.I \circ \top\!\!\top \\
= & \qquad \{ \;\text{domains: } H.I \circ \top\!\!\top \;=\; H.\top\!\!\top \circ \top\!\!\top \; \} \\
 & zip_{F,G} \circ F.G.\top\!\!\top \circ \top\!\!\top \\
= & \qquad \{ \;\text{domains: } H.I \circ \top\!\!\top \;=\; H.\top\!\!\top \circ \top\!\!\top \; \} \\
 & zip_{F,G} \circ F.G.\top\!\!\top \circ F.\top\!\!\top \circ \top\!\!\top \\
= & \qquad \{ \;\text{relator fusion, (21)}\; \} \\
 & zip_{F,G} \circ F.\hat{G}.\top\!\!\top \circ \top\!\!\top \\
= & \qquad \{ \;(13)\; \} \\
 & \hat{G}.F.\top\!\!\top \circ \top\!\!\top \quad .
\end{array}$$

Property (24) states that $zip_{F,G}$ constructs *all* G-structures of (F-structures of the same shape).

The above equations are valid in the case that F and G are both unary relators. Some care is needed in "lifting" these equations to binary relators. We state the requirements only without justification: If F is a unary relator and $\oplus$ and $\otimes$ are binary relators then $\hat{\oplus}$ is required to satisfy

$$(25) \qquad R\hat{\oplus}S \;=\; R\oplus S \circ I\hat{\oplus}I \quad \text{for all } R,S \quad ,$$

$$(26) \qquad \top\!\!\top\hat{\oplus}\top\!\!\top \;=\; \top\!\!\top\oplus\top\!\!\top \circ \top\!\!\top \quad ,$$

$$(27) \qquad f\hat{\oplus}f \;=\; I\hat{\oplus}I \circ f \quad \text{for all imps } f$$

(and, of course, similarly for $\otimes$).

The requirements for $zip_{F,\oplus}$ are:

$$(28) \qquad zip_{F,\oplus} \circ F.(R\hat{\oplus}S) \;=\; F.R \,\hat{\oplus}\, F.S \quad ,$$

$$(29) \qquad F.R \oplus F.S \circ zip_{F,\oplus} \;=\; zip_{F,\oplus} \circ F.(R\oplus S) \quad ,$$

$$(30)\qquad F.I \oplus F.I \circ zip_{F,\oplus} \;=\; zip_{F,\oplus} \;=\; zip_{F,\oplus} \circ F.(I{\oplus}I) \;\; .$$

The requirements for $zip_{\oplus,F}$ are:

$$(31)\qquad zip_{\oplus,F} \circ \hat{F}.R \oplus \hat{F}.S \;=\; \hat{F}.(R{\oplus}S) \;\; ,$$

$$(32)\qquad F.(R{\oplus}S) \circ zip_{\oplus,F} \;=\; zip_{\oplus,F} \circ F.R \oplus F.S \;\; ,$$

$$(33)\qquad F.(I{\oplus}I) \circ zip_{\oplus,F} \;=\; zip_{\oplus,F} \;=\; zip_{\oplus,F} \circ F.I \oplus F.I \;\; .$$

And, finally, the requirements for $zip_{\otimes,\oplus}$ are

$$(34)\qquad zip_{\otimes,\oplus} \circ (R\hat{\oplus}S) \otimes (T\hat{\oplus}U) \;=\; (R{\otimes}T) \,\hat{\oplus}\, (S{\otimes}U) \;\; ,$$

$$(35)\qquad (R \oplus S) \otimes (T \oplus U) \circ zip_{\oplus,\otimes} \;=\; zip_{\oplus,\otimes} \circ (R \otimes T) \oplus (S \otimes U) \;\; ,$$

$$(36)\qquad (I \oplus I) \otimes (I \oplus I) \circ zip_{\oplus,\otimes} \;=\; zip_{\oplus,\otimes} \;\; \wedge$$

$$zip_{\oplus,\otimes} \;=\; zip_{\oplus,\otimes} \circ (I \otimes I) \oplus (I \otimes I) \;\; .$$

(Note the switch in order of the variables R, S, T and U in (34) and (35)).

Exercise 37 Verify that for all R, S, T, U,

$$(38)\qquad F.R \circ \hat{F}.S \;=\; \hat{F}.(R \circ S)$$

and

$$(39)\qquad R{\oplus}S \circ T\hat{\oplus}U \;=\; (R \circ T) \,\hat{\oplus}\, (S \circ U) \;\; .$$

These two properties are used often and will be referred to by the name *generator fusion.*

□

5 The Zip-a-dee-doo-dah Theorem

We are now ready to formulate a general theorem about the existence of zips. We call the theorem the zip-a-dee-doo-dah theorem after the song "Zip-a-dee-doo-dah, zip-a-dee-ay" because it seemed such a wonderful day when the theorem was first discovered (at least to us anyway)! The inspiration for the theorem came from Fokkinga's Ph.D. thesis [1, chapter 4, p 90 onwards]. Fokkinga considers only one particular case — the construction of a function that transposes a stream of lists into a list of streams — but it was clear from his calculations that they could be made much more general if the step were taken to a higher level of abstraction. (Fokkinga was himself unable to take this step because he was not prepared to admit relations to his calculus and was therefore unable to formulate the notions of generator and co-generator.)

Polynomial relators in the spec calculus fit into three categories. The primitive relators to wit sum $(+)$, product $(\times)$ and the constant relator $\mathbb{1}\bullet$ (where $(\mathbb{1}\bullet).R \ = \ \mathbb{1}$); the relators formed by composing existing relators; and the map relators. We define a subclass of the polynomial relators:

Definition 40 The class ζ is defined to be the smallest class of relators such that

(a) the binary relators $+$ and $\times$ are elements of ζ,

(b) a composition of relators each of which is in ζ is also in ζ,

(c) if $\ominus$ is a binary relator in ζ then the map relator $\dagger$ defined by $\dagger.R = \mu(R\ominus)$ is in ζ.

$\square$

Note that ζ is a subclass of the bottom-strict polynomial relators. Furthermore, all relatos of ζ are $\sqcap$-junctive.

In this paper we are not precise about what we mean by a "composition of relators". The problem is that it is necessary to compose relators of different arities in order to obtain a worthwhile class of relators and to do this we need to go at least one step further in the level of abstraction at which we conduct our arguments.

The class ζ will be the class of relators under consideration in this paper. We construct for all ordered pairs (F, G) of relators such that F is strict and $\sqcap$-junctive and G is an element of ζ a spec $zip_{F,G}$ that witnesses the "zippability" of (F, G). Furthermore we show that if both F and G are in ζ then they are commuting relators. Specifically, we claim:

Theorem 41 (Zip-a-dee-doo-dah) Suppose F is a bottom-strict, $\sqcap$-junctive relator and suppose G is a relator in the class ζ. Then there is a spec $zip_{F,G}$ such that the ordered pair (F, G) is zippable by $zip_{F,G}$. Furthermore, if F also belongs to the class ζ then there is a spec $zip_{G,F}$ such that the ordered pair (G, F) is zippable by $zip_{G,F}$ and $zip_{F,G} = (zip_{G,F})^{\cup}$. (Thus, by definition, F and G are commuting relators.)
□

It should not be supposed that the class ζ demarcates all commuting relators. We have strong reasons to suspect that what we call "zips" appear under other names in other research publications (in particular, under the name "cross operator" in publications of de Moor and Bird) and a worthwhile research problem would be to extend the class yet further.

The proof of the theorem is by induction on the structure of the relators in the class ζ. This leads to a rather long proof comprising in total 26 components! These components are: 4×4 lemmas establishing (13)–(16) for each of $+$, $\times$, a composition of relators and a map relator, and $10 = 4+3+2+1$ lemmas establishing (18) for each combination of F and G among these same four possibilities. In view of its length we have subdivided the proof among several sections (and subsections thereof).

First, we show how to construct generators for each subclass of relators comprising the class ζ. This we do in section 6. Then we show how to construct zips $zip_{F,G}$ for each $\sqcap$-junctive relator F and each relator G in each subclass of the class ζ. (Both these constructions are inductive over the structure of relators in ζ.) This part of the argument is split into two steps. In section 7 we construct "candidate" zips. The construction of "candidate" zips is dictated solely by the desire to meet property (13) in the definition of "zippable". Subsequently, in section 8, we verify that the construction we give simultaneously satisfies the remaining clauses in the definition of "zippable". This completes the first 4×4 elements of the proof. In section 9 we complete the proof by showing that that the ten different pairs of zips are indeed inverses of each other. (although several proofs are left to the reader as an exercise.)

To show that this amount of effort is indeed worthwhile we conclude the paper with some delightful and surprising consequences of the existence of zips.

6 Constructing Generators

Our first task is to construct generators for all relators in the class ζ. This we do in this section.

6.1 Product and Sum

We have already seen the product generator — it is split. Properties (25) to (27) have already been discussed in exercise 4. The generator for sum can be deduced as follows. Consider (25). For product this is the fusion property

$$R \times S \circ I \vartriangle I \quad = \quad R \vartriangle S$$

For sum we require that

$$R + S \circ I \hat{+} I \quad = \quad R \hat{+} S$$

Recalling the fusion property of sum:

$$T \triangledown U \circ P + Q \quad = \quad (T \circ P) \triangledown (U \circ Q)$$

In particular,

$$I \triangledown I \circ P + Q \quad = \quad P \triangledown Q$$

we are led to guess at $(I \triangledown I)^{\cup}$ for $I \hat{+} I$. It then follows that

$$\begin{array}{ll} & R \hat{+} S \\ = & \qquad \{ \ (25) \ \} \\ & R + S \circ (I \triangledown I)^{\cup} \\ = & \qquad \{ \ \text{reverse} \ \} \\ & (I \triangledown I \circ R^{\cup} + S^{\cup})^{\cup} \\ = & \qquad \{ \ \text{junc-sum fusion} \ \} \\ & (R^{\cup} \triangledown S^{\cup})^{\cup} \end{array}$$

We normally write $R \blacktriangledown S$ instead of $R \hat{+} S$ (it looks nicer) and pronounce it co-junc. It is thus a sort of conjugate of junc. The other two properties (26) and (27) are left as exercises. By analogy we write $R \blacktriangle S$ for $(R^{\cup} \vartriangle S^{\cup})^{\cup}$.

6.2 Composition of Relators

In this section we face the problem of our not having formalised the notion of the composition of relators of different arities. The (unsubstantiated) claim we make, however, is that all calculations with unary relators can be "lifted" without difficulty to calculations with relators of higher arity. For two unary relators, F and G, the rule is that the generator of their composition is the composition of their generators. Thus,

$$(42) \qquad (F \bullet G)\hat{} \;=\; \hat{F} \bullet \hat{G} \quad .$$

We leave to the reader the straightforward task of verifying that this definition of $(F \bullet G)\hat{}$ meets the specification of the generator of $F \bullet G$.

For relators of higher arities we apply exactly the same rule (the generator of a composition is the composition of the generators). For example, the relator

$$R, S, T \longmapsto R + (S \times T)$$

has generator

$$R, S, T \longmapsto R \blacktriangledown (S \vartriangle T) \quad .$$

Try verifying that this function meets its specification and you will see what we mean by the claim that all calculations with unary relators can be "lifted" without difficulty to calculations with relators of higher arity. Nevertheless, this is no excuse for lack of a proper analysis, and we offer our sincere apologies. In future versions of the paper we intend to rectify the shortcoming.

6.3 Map Relators

We start with an exercises as a short introduction to so-called *anamorphisms*. To be honest, this theory should have been included in "A Relational Theory of Datatypes.".

Exercise 43 Many functions on datatypes involve two phases. In the first phase a data structure is generated and in the second phase the data structure is collapsed. The second phase is expressed by a catamorphism, the first by a so-called *anamorphism*. If F is a relator and R is a spec, the anamorphism $[\![F;\ R]\!]$ is defined by

$$(44) \qquad [\![F;\ R]\!] \quad = \quad (\![F;\ R^{\cup}]\!)^{\cup}$$

Establish the following properties of anamorphisms:

Unique extension property:

$$(45) \qquad X = [\![F;\ R]\!] \quad \equiv \quad X = \mu F \circ F.X \circ R$$

Computation rule:

$$(46) \qquad [\![F;\ R]\!] \quad = \quad F.[\![F;\ R]\!] \circ R$$

Fusion law:

$$(47) \qquad [\![F;\ R]\!] \circ S = [\![F;\ T]\!] \quad \Leftarrow \quad R \circ S = F.S \circ T$$

Domain law:

$$(48) \qquad [\![F;\ R]\!] \quad = \quad \mu F \circ [\![F;\ R]\!]$$

Identity law:

$$(49) \qquad \mu F \quad = \quad [\![F;\ \mu F]\!]$$

Show that $(\![F;\ R]\!) \circ [\![F;\ S]\!]$ satisfies the equation

$$X :: \quad X \quad = \quad R \circ F.X \circ S$$

(It is in fact the least solution)
□

(Note: the name "anamorphism" is due to Erik Meijer.)

Next we continue with the generator of the map relator. For map relators it is also possible to use elementary reasoning to identify a candidate generator and then verify that (20)–(22) are satisfied.

Suppose $\ominus$ is a binary relator. In section "Parameterised Types" of "A Relational Theory of Datatypes" the map relator $\dagger$ is defined by

$$(50) \qquad \dagger.R \quad = \quad \mu(R\ominus) \quad .$$

Note that $\mu(R\ominus)$ is the least solution of

$$X :: \quad X \quad = \quad R \ominus X \quad = \quad (I\ominus).X \circ R \ominus I \quad = \quad R \ominus I \circ (I\ominus).X$$

So, $\mu(R\ominus)$ is a $I\ominus$ cata– and anamorphisms:

$$(51) \qquad \dagger.R \quad = \quad \mu(R\ominus) \quad = \quad (\!|I\ominus;\ R\ominus I|\!) \quad = \quad [\![I\ominus;\ R\ominus I]\!]$$

We assume that $\ominus$ has generator $\hat{\ominus}$. Now the equation

$$\dagger.R \quad = \quad R \ominus (\dagger.R)$$

suggests that we require that

$$\hat{\dagger}.R \quad = \quad R\hat{\ominus}(\hat{\dagger}.R)$$

and, to make life simple,

$$\hat{\dagger}.R \quad = \quad \mu(R\hat{\ominus}) \quad .$$

(Note that $R\hat{\ominus}$ is monotonic since, by definition, $\hat{\ominus}$ is.) Since $\mu(R\hat{\ominus})$ is the least solution of

$$X :: \quad X \quad = \quad R\hat{\ominus}X \quad = \quad (I\ominus).X \circ R\hat{\ominus}I$$

it follows that $\mu(R\hat{\ominus})$ is an $I\ominus$ anamorphism:

$$(52) \qquad \hat{\dagger}.R \quad = \quad \mu(R\hat{\ominus}) \quad = \quad [\![I\ominus;\ R\,\hat{\ominus}\,I]\!] \quad .$$

Note that $\hat{\dagger}.R$ is in general not a catamorphism.

We verify (20) by applying fusion for anamorphisms:

$$
\begin{array}{ll}
 & \hat{\dagger}.R \quad = \quad \dagger.R \circ \hat{\dagger}.I \\
\equiv & \qquad \{ \ (52), \text{anamorphisms} \ \} \\
 & [\![I\ominus;\ R\hat{\ominus}I]\!] \quad = \quad [\![I\ominus;\ R\ominus I]\!] \circ \hat{\dagger}.I \\
\Leftarrow & \qquad \{ \ \text{fusion} \ \} \\
 & I \ominus \hat{\dagger}.I \circ R\hat{\ominus}I \quad = \quad R \ominus I \circ \hat{\dagger}.I \\
\equiv & \qquad \{ \ \text{generator fusion, } \hat{\dagger}.I \quad = \quad I\hat{\ominus}(\hat{\dagger}.I) \ \} \\
 & R \ominus I \circ I\hat{\ominus}\hat{\dagger}.I \quad = \quad R \ominus I \circ I\hat{\ominus}\hat{\dagger}.I \\
\equiv & \\
 & \mathit{true} \quad .
\end{array}
$$

Property (21) we verify by

$$
\begin{array}{ll}
 & \hat{\dagger}.\top\!\top \;=\; \dagger.\top\!\top \circ \top\!\top \\
\equiv & \qquad \{\ (52),\ \text{anamorphisms}\ \} \\
 & [\![I\ominus;\ \top\!\top \hat{\ominus} I]\!] \;=\; [\![I\ominus;\ \top\!\top \ominus I]\!] \circ \top\!\top \\
\Leftarrow & \qquad \{\ \text{fusion}\ \} \\
 & I \ominus \top\!\top \circ \top\!\top \hat{\ominus} I \;=\; \top\!\top \ominus I \circ \top\!\top \\
\equiv & \qquad \{\ \text{generator fusion, domains}\ \} \\
 & \top\!\top \hat{\ominus} \top\!\top \;=\; \top\!\top \ominus \top\!\top \circ \top\!\top \\
\equiv & \qquad \{\ (26)\ \} \\
 & \mathit{true}\ .
\end{array}
$$

Property (22) is equally straightforward and is left as an exercise.

6.4 Summary and an Example

Let us summarise the results of this section. The generators of product and sum are $\vartriangle$ and $\blacktriangledown$, respectively, where

$$R \blacktriangledown S \;=\; (R^{\cup} \triangledown S^{\cup})^{\cup}\ .$$

The generator of a composition of relators is the composition of the generators. Finally the generator of the map relator is $R \mapsto [\![I\ominus;\ R\hat{\ominus}I]\!]$.

Combining these results we get the generator of non-empty lists. Specifically, the relator PL (positive, i.e, non-empty, lists) is given by

$$PL.R \;=\; \mu(R\ominus)$$

where $R \ominus S \;=\; R + R \times S$.
The generator associated with $\ominus$ is thus

$$R\hat{\ominus}S \;=\; R \blacktriangledown (R \vartriangle S)$$

and that associated with PL is

$$\hat{PL}.R \;=\; [\![I\ominus;\ R \blacktriangledown (R \vartriangle I)]\!]\ .$$

Exercise 53 Consider the following

$$\begin{array}{lcl} Ash.R & = & \mu(X \mapsto R + (X \times R) \times X) \\ Box.R & = & \mu(X \mapsto PL.X \times R) \\ Elm.R & = & PL.Box.R \\ Fir.R & = & \mu(X \mapsto R + Ash.X) \end{array}$$

What are their generators? How would you describe an ash of elms all of the same shape?
□

7 Constructing Candidate Zips

We are ready to embark on the task of constructing $zip_{F,G}$ for various combinations of relators F and G. We concentrate initially on meeting (13), i.e. on constructing a spec $zip_{F,G}$ such that

$$(54) \qquad zip_{F,G} \circ F.\hat{G}.R \; = \; \hat{G}.F.R$$

for all specs R. The specs we construct are, at this point in time, *candidate* zips. In the next section we vet them against the three other clauses in the specification of a zip.

The construction is by induction on the structure of relators in the class of bottom-strict polynomial relators. That is, we show how to construct $zip_{F,+}$ and $zip_{F,\times}$ for arbitrary relator F. We show how to construct $zip_{F,G}$ for G a composition of relators by composing the zips for the components of G. And we show how to construct $zip_{F,\dagger}$ for a map relator † given that $zip_{F,\ominus}$ has already been constructed.

7.1 Zipping Disjoint Sums

Because of the problematic behaviour of split with respect to non-determinacy it is significantly more difficult to construct $zip_{F,\times}$ than it is to construct $zip_{F,+}$. (It will turn out that we need to restrict F in the former case.) We therefore begin with $zip_{F,+}$.

Rewriting (54) taking into account the fact that $+$ is a binary relator and $\hat{+} \; = \; \blacktriangledown$ we are required to construct $zip_{F,+}$ satisfying

$$(55) \qquad zip_{F,+} \circ F.(R \blacktriangledown S) \; = \; F.R \; \blacktriangledown \; F.S$$

for all specs R and S.

Let us abbreviate $zip_{F,+}$ by $\alpha_\cup$ and at the same time replace $\blacktriangledown$ in (55) by $\triangledown$. Then

$$
\begin{array}{ll}
 & \forall(R,S :: \ zip_{F,+} \circ F.(R \blacktriangledown S) \ = \ F.R \blacktriangledown F.S) \\
\equiv & \qquad \{ \ \alpha \ = \ (zip_{F,+})_\cup \ \} \\
 & \forall(R,S :: \ \alpha_\cup \circ F.(R \blacktriangledown S) \ = \ F.R \blacktriangledown F.S) \\
\equiv & \qquad \{ \ \text{reverse} \ \} \\
 & \forall(R,S :: \ F.(R \blacktriangledown S)_\cup \circ \alpha \ = \ (F.R \blacktriangledown F.S)_\cup) \\
\equiv & \qquad \{ \ (X \blacktriangledown Y)_\cup \ = \ X_\cup \triangledown Y_\cup \ \} \\
 & \forall(R,S :: \ F.(R_\cup \triangledown S_\cup) \circ \alpha \ = \ F.R_\cup \triangledown F.S_\cup) \\
\equiv & \qquad \{ \ \text{dummy change: } R,S \mapsto R_\cup, S_\cup \ \} \\
 & \forall(R,S :: \ F.(R \triangledown S) \circ \alpha \ = \ F.R \triangledown F.S)
\end{array}
$$

So we are required to construct a *co-imp* α such that

$$
(56) \qquad F.(R \triangledown S) \circ \alpha \ = \ F.R \triangledown F.S
$$

Remark The preceeding calculation is quite unnecessary if you are as familiar with the rules for cojunc as you are with those for junc. In that case $zip_{F,+}$ can be constructed directly from (55). We assume, however, that this is not the case. *End of Remark*

The construction involves beginning with the left side of (56) and trying to choose the shape of α so that it can be transformed to the right side. The choices made are marked in the calculation by a bullet.

$$
\begin{array}{lll}
 & F.(R \triangledown S) \circ \alpha & \\
= & \qquad \{ & \text{The right side of (56) is a junc. By choosing } \alpha \text{ to be} \\
 & & \text{a junc we can exploit spec-junc fusion.} \\
 & \qquad \bullet & \alpha \ = \ \beta \triangledown \gamma \ \} \\
 & F.(R \triangledown S) \circ \beta \triangledown \gamma & \\
= & \qquad \{ & \text{spec-junc fusion } \} \\
 & (F.(R \triangledown S) \circ \beta) \triangledown (F.(R \triangledown S) \circ \gamma) & \\
= & \qquad \{ & \text{The two arguments of junc in the right side of (56)} \\
 & & \text{are of the form } F.X. \text{ By choosing } \beta \text{ and } \gamma \text{ to have} \\
 & & \text{this form we can exploit the fact that } F \text{ is a relator.} \\
 & \qquad \bullet & \beta \ = \ F.\delta \ , \ \gamma \ = \ F.\varepsilon \ \} \\
 & F.(R \triangledown S \circ \delta) \triangledown F.(R \triangledown S \circ \varepsilon) & \\
= & \qquad \{ & \text{This last step hardly needs explanation. Compare}
\end{array}
$$

what we have with the rhs of (56).

- $\delta \;=\; \hookrightarrow \,,\, \varepsilon \;=\; \hookleftarrow$
  computation rule }

$F.R \triangledown F.S$.

In this way we have constructed

$$(57) \qquad \alpha \;=\; F.\!\hookrightarrow \triangledown F.\!\hookleftarrow$$

and shown that it satisfies (56). We conclude that $\alpha^{\cup}$ satisfies the requirement (55) on $zip_{F,+}$. Reintroducing $\blacktriangledown$ we have

$$(58) \qquad zip_{F,+} \;=\; F.\!\hookrightarrow^{\cup} \blacktriangledown F.\!\hookleftarrow^{\cup} \;.$$

7.2 Zipping Products

Next we consider the construction of $zip_{F,\times}$. Earlier we said that its construction is substantially harder than that of $zip_{F,+}$. The problem is that the construction we have just given does not dualise to product since the second step used spec-junc fusion whereas its dual — split-spec fusion — does not hold in general. (An example of swings and roundabouts: admitting relations into the calculus gives a very compact definition of *zip* but leads to difficulties in its construction!) We can however dualise $zip_{F,+}$ and speculate that it is $zip_{F,\times}$! In other words we conjecture that, for all relators F and all R, S,

$$(F.\!\ll \vartriangle F.\!\gg) \circ F.(R \vartriangle S) \;=\; F.R \vartriangle F.S \;.$$

I.e. the candidate zip for products is

$$(59) \qquad zip_{F,\times} \;=\; F.\!\ll \vartriangle F.\!\gg \;.$$

Unfortunately the conjecture doesn't hold in its entirety. It *does* hold, however, for all $\sqcap$-junctive relators. I.e. we have

$$(60) \qquad \begin{array}{ll} & F \text{ is } \sqcap\text{-junctive} \\ \Rightarrow & \forall(R,S :: \;(F.\!\ll \vartriangle F.\!\gg) \circ F.(R \vartriangle S) \;=\; F.R \vartriangle F.S) \end{array}$$

Moreover, all relators in the class in which we are interested are $\sqcap$-junctive.

The (longish) verification of (60) completes this section. You are advised to skip to the next section on a first reading.

We first establish a theorem that is important in its own right. Specifically

(61)
$$
\begin{array}{ll}
 & F \text{ is } \sqcap\text{-junctive} \\
\equiv & \forall(R,S :: \;\; F.(R \vartriangle S) \;\; = \;\; (F.\!\ll \vartriangle F.\!\gg)^{\cup} \circ F.R \vartriangle F.S)
\end{array}
$$

The proof is by a ping-pong argument. First, implication:

$$
\begin{array}{ll}
 & F.(R \vartriangle S) \\
= & \quad \{ \text{ definition } \vartriangle, \;\; F \text{ is } \sqcap\text{-junctive } \} \\
 & F.(\ll^{\cup} \circ R) \sqcap F.(\gg^{\cup} \circ S) \\
= & \quad \{ \text{ relator.} F \; \} \\
 & (F.\!\ll^{\cup} \circ F.R) \sqcap (F.\!\gg^{\cup} \circ F.S) \\
= & \quad \{ \text{ axiom } \} \\
 & (F.\!\ll \vartriangle F.\!\gg)^{\cup} \circ F.R \vartriangle F.S
\end{array}
$$

Second, “follows-from”:

$$
\begin{array}{ll}
 & F.(R \sqcap S) \;\; = \;\; F.R \sqcap F.S \\
\equiv & \quad \{ \text{ axiom } \} \\
 & F.((I \vartriangle I)^{\cup} \circ R \vartriangle S) \;\; = \;\; (F.I \vartriangle F.I)^{\cup} \circ F.R \vartriangle R.S \\
\equiv & \quad \{ \text{ relator.} F \\
 & \qquad F.(R \vartriangle S) \;\; = \;\; (F.\!\ll \vartriangle F.\!\gg)^{\cup} \circ F.R \vartriangle F.S \\
 & F.(I \vartriangle I)^{\cup} \circ (F.\!\ll \vartriangle F.\!\gg)^{\cup} \circ F.R \vartriangle F.S \\
 & \qquad = \;\; (F.I \vartriangle F.I)^{\cup} \circ F.R \vartriangle F.S \\
\Leftarrow & \quad \{ \text{ Leibniz } \} \\
 & F.(I \vartriangle I)^{\cup} \circ (F.\!\ll \vartriangle F.\!\gg)^{\cup} \;\; = \;\; (F.I \vartriangle F.I)^{\cup} \\
\equiv & \quad \{ \text{ reverse } \} \\
 & F.\!\ll \vartriangle F.\!\gg \circ F.(I \vartriangle I) \;\; = \;\; F.I \vartriangle F.I
\end{array}
$$

Continuing with the lhs:

$$
\begin{array}{ll}
 & F.\!\ll \vartriangle F.\!\gg \circ F.(I \vartriangle I) \\
= & \quad \{ \text{ split-imp fusion, } F.(I \vartriangle I) \text{ is an imp } \} \\
 & (F.\!\ll \circ F.(I \vartriangle I)) \vartriangle (F.\!\gg \circ F.(I \vartriangle I)) \\
= & \quad \{ \; F \text{ is a relator, computation rule } \} \\
 & F.I \vartriangle F.I
\end{array}
$$

This completes the proof of (61).

Comparing the right side of (61)

$$F.(R \vartriangle S) \quad = \quad (F.\ll \vartriangle F.\gg)^{\cup} \circ F.R \vartriangle F.S$$

with the desired property

$$F.\ll \vartriangle F.\gg \circ F.(R \vartriangle S) \quad = \quad F.R \vartriangle F.S$$

it is clear that what is required is

$$(62) \qquad F.\ll \vartriangle F.\gg \circ (F.\ll \vartriangle F.\gg)^{\cup} \circ F.R \vartriangle F.S \quad = \quad F.R \vartriangle F.S \quad .$$

Since $F.\ll \vartriangle F.\gg$ is an imp,

$$(63) \qquad F.\ll \vartriangle F.\gg \circ (F.\ll \vartriangle F.\gg)^{\cup} \quad = \quad (F.\ll \vartriangle F.\gg)<$$

so we can use the domain laws to establish (62).
In fact we calculate

$$(64) \qquad (F.\ll \vartriangle F.\gg)< \quad = \quad (F.\top\!\!\top \vartriangle F.\top\!\!\top)<$$

as follows

$$
\begin{array}{ll}
 & (F.\ll \vartriangle F.\gg)< \\
= & \qquad \{ \text{ domain trading } \} \\
 & ((F.\ll \circ F.\gg^{\cup}) \vartriangle I)< \\
= & \qquad \{ \text{ relator.}F \text{ , } \ll \circ \gg^{\cup} \;=\; \top\!\!\top \text{ } \} \\
 & (F.\top\!\!\top \vartriangle I)< \\
= & \qquad \{ \; F.\top\!\!\top \;=\; F.\top\!\!\top \circ F.\top\!\!\top^{\cup} \text{, domain trading } \} \\
 & (F.\top\!\!\top \vartriangle F.\top\!\!\top)< \quad .
\end{array}
$$

Now (62) is straightforward:

$$
\begin{array}{ll}
 & (62) \\
\equiv & \qquad \{ \text{ (63), domains } \} \\
 & (F.R \vartriangle F.S)< \;\sqsubseteq\; (F.\top\!\!\top \vartriangle F.\top\!\!\top)< \\
\equiv & \qquad \{ \; R \sqsubseteq \top\!\!\top \text{ , } S \sqsubseteq \top\!\!\top \text{ , monotonicity } \} \\
 & \mathit{true} \quad .
\end{array}
$$

□

Exercise 65 Formulae (58) and (59) predict formulae for $zip_{+,+}$ and $zip_{\times,\times}$. What are these formulae? Verify that

(a) $\quad zip_{+,+} \;=\; (zip_{+,+})^{\cup}$

(b) $\quad zip_{\times,\times} \;=\; (zip_{\times,\times})^{\cup}$

(In (b) you may wish to use the fact that the projections $\ll$ and $\gg$ are imps.)

□

Exercise 66 There is a commonality in the calculations with sum and product which this exercise captures and generalises. With the aid of this exercise one is able to handle both simultaneously rather than doing two sets of calculations, one for sum and the other for product.

Assume that G is a relator and α is a spec with $G.I \;=\; \hat{G}.\alpha$. Prove that, for all relators F, if there exists a γ such that:

(67) $\quad \gamma \circ F.\hat{G}.R \;=\; \hat{G}.F.R \;\;,$

(68) $\quad \gamma \circ F.G.I \;=\; \gamma \;\;.$

then we have that:

(69) $\quad \gamma \;=\; \hat{G}.F.\alpha \;=\; (\check{G}.F.\alpha^{\cup})^{\cup} \;\;.$

Prove also, using the same assumption about relator G, that if

(70) $\quad \hat{G}.F.\alpha \circ F.\hat{G}.R \;=\; \hat{G}.F.R \;\;.$

then we have

(71) $\quad G.F.R \circ \hat{G}.F.\alpha \;=\; \hat{G}.F.\alpha \circ F.G.R \;\;,$

(72) $\quad G.F.I \circ \hat{G}.F.\alpha \;=\; \hat{G}.F.\alpha \;=\; \hat{G}.F.\alpha \circ F.G.I \;\;.$

From (69) it follows that if there exists a $zip_{F,G}$ it must be equal to $\hat{G}.F.\alpha$ and it is therefore unique. Furthermore, from (71) and (72 it follows that if the only possible candidate for $zip_{F,G}$, i.e. $\hat{G}.F.\alpha$, satisfies (13) we get the naturality property (14) and domain property (15) for free. Both facts we will further exploit in the remainder part of this paper.

The relevance of this theorem to sum and product is seen by observing that

(73) $\quad I + I \;=\; \hookrightarrow^{\cup} \blacktriangledown \hookleftarrow^{\cup} \;=\; (\hookrightarrow \triangledown \hookleftarrow)^{\cup}$

(74) $\quad I \times I \;=\; \ll \vartriangle \gg \;=\; (\ll^{\cup} \blacktriangle \gg^{\cup})^{\cup} \;\;.$

You should compare (73) and (74) with (69) noting that $\blacktriangledown \;=\; \hat{+}$ and $\vartriangle \;=\; \hat{\times}$.

□

7.3 Zipping Compositions

Because we haven't been specific about how relators are composed to form new relators we can't formulate a general theorem about how to construct the *zip* when one or other of the argument relators is a composition. Instead we shall illustrate the method with an example (which happens to be relevant to the construction of the *zip* function from pairs of lists to lists of pairs).

In the exercises we ask you to derive the zip function for a composition of two unary relators. As always, our claim is that this construction can always be lifted to the case of relators of higher arity.

Let G be the ternary relator given by $G.(R,S,T) \;=\; R+(S\times T)$. Let F be an arbitrary $\sqcap$-junctive relator. Then we construct $zip_{F,G}$.

Recall that $\hat{G}.(R,S,T) \;=\; R \blacktriangledown (S \vartriangle T)$. We construct α such that

$$\alpha \circ F.\hat{G}.(R,S,T) \;=\; \hat{G}.(F.R, F.S, F.T) \;.$$

i.e.

(75) $$\alpha \circ F.(R \blacktriangledown (S \vartriangle T)) \;=\; F.R \blacktriangledown (F.S \vartriangle F.T)$$

We calculate

$$\begin{array}{cl}
 & \alpha \circ F.(R \blacktriangledown (S \vartriangle T)) \\
= & \quad \{ \;\bullet\; \alpha \;=\; \beta \circ zip_{F,+},\ (54) \;\} \\
 & \beta \circ F.R \blacktriangledown F.(S \vartriangle T) \\
= & \quad \{ \;\bullet\; \beta \;=\; I+\gamma,\ +\text{-}\blacktriangledown \text{ fusion} \;\} \\
 & F.R \blacktriangledown (\gamma \circ F.(S \vartriangle T)) \\
= & \quad \{ \;\bullet\; \gamma \;=\; zip_{F,\times} \;\} \\
 & F.R \blacktriangledown (F.S \vartriangle F.T) \;.
\end{array}$$

We have thus constructed

$$\alpha \;=\; I + zip_{F,\times} \circ zip_{F,+} \;.$$

Filling in the details

(76) $$\alpha = I + (F.\!\ll \vartriangle F.\!\gg) \circ (F.\!\hookleftarrow \triangledown F.\!\hookrightarrow)^{\cup} \;.$$

Exercise 77 Suppose G and H are both unary relators. Denote their composition by $G \bullet H$. (Thus $(G \bullet H).R \;=\; G.(H.R))$. Let F be a relator and suppose $zip_{F,G}$ and $zip_{F,H}$ both exist. Construct a candidate for $zip_{F,G\bullet H}$ in terms of these two which satisfies (13).
□

7.4 Zipping Map Relators

The final stage in this analysis of zips is their construction for map relators. At the end of this section we will be able to exhibit the zip with which this whole discussion began — that from pairs of (non-empty) lists to lists of pairs. This we do explicitly. But there is much more we can do. Just by instantiating the results we have one can construct (candidate) "zips" for the complete family of relators we have been discussing. Examples include zipping (non-empty) trees of roses to form roses of trees — for all sorts of trees ("ashes", "elms" etc.) —, a disjoint sum of roses to form a rose of disjoint sums, and lots of other exotic and nameless species of data structures.

Let $\dagger$ be the map relator defined by $\dagger.R \;=\; \mu(R\ominus)$ where $\ominus$ is a binary relator. We consider the construction of $zip_{F,\dagger}$ for arbitrary relator F. The construction is inductive in the sense that we assume the existence of $zip_{F,\ominus}$ and construct $zip_{F,\dagger}$ in terms of it.

The task is to compute $\alpha \;=\; zip_{F,\dagger}$ such that

$$\alpha \circ F.\hat{\dagger}.R \;=\; \hat{\dagger}.F.R$$

for all specs R . Recalling that

$$(78) \qquad \hat{\dagger}.S \;=\; [\![I\ominus;\; S\hat{\ominus}I]\!]$$

we calculate α as follows

$$
\begin{array}{ll}
 & \alpha \circ F.\hat{\dagger}.R \;=\; \hat{\dagger}.F.R \\
\equiv & \qquad \{ \;\; (78) \;\; \} \\
 & \alpha \circ F.\hat{\dagger}.R \;=\; [\![F.R\hat{\ominus}I]\!] \\
\equiv & \qquad \{ \;\; \bullet \quad \alpha \;=\; [\![\beta]\!], \quad \text{heading for fusion} \;\; \} \\
 & [\![\beta]\!] \circ F.\hat{\dagger}.R \;=\; [\![F.R\hat{\ominus}I]\!] \\
\Leftarrow & \qquad \{ \;\; \text{anamorphism fusion} \;\; \} \\
 & \beta \circ F.\hat{\dagger}.R \;=\; I \ominus F.\hat{\dagger}.R \circ F.R \hat{\ominus}\; I \\
\equiv & \qquad \{ \;\; \text{generator fusion, (39)} \;\; \} \\
 & \beta \circ F.\hat{\dagger}.R \;=\; F.R \hat{\ominus} F.\hat{\dagger}.R \\
\equiv & \qquad \{ \;\; \hat{\dagger}.S \;=\; S \hat{\ominus} \hat{\dagger}.S \;\; \} \\
 & \beta \circ F.(R \hat{\ominus} \hat{\dagger}.R) \;=\; F.R \hat{\ominus} F.\hat{\dagger}.R \\
\Leftarrow & \qquad \{ \;\; \text{calculus} \;\; \} \\
 & \forall (S,T :: \;\; \beta \circ F.(S\hat{\ominus}T) \;=\; F.S \hat{\ominus} F.T)
\end{array}
$$

$$\begin{array}{ll} \Leftarrow & \{ \text{ definition } \} \\ & \beta \ = \ zip_{F,\ominus} \ . \end{array}$$

We have thus constructed $\alpha = [\![I\ominus;\ zip_{F,\ominus}]\!]$
I.e.

$$(79) \qquad zip_{F,\dagger} \ = \ [\![I\ominus;\ zip_{F,\ominus}]\!] \ .$$

Let us conclude by applying (78) to non-empty lists. Recalling that

$$\begin{array}{lrcl} & PL.R & = & \mu(R\ominus) \\ \text{where} & R\ominus S & = & R+(R\times S) \end{array}$$

equation (79) says that $zip_{F,PL} \ = \ [\![I\ominus;\ \beta]\!]$ where $\beta \ = \ zip_{F,\ominus}$. The latter has already been calculated. In (75) and (76) we recorded that for all R, S and T

$$\begin{array}{ll} & I+(F.\ll \vartriangle F.\gg) \circ (F\hookrightarrow \triangledown F.\hookleftarrow)^{\cup} \circ F.(R \blacktriangledown (S \vartriangle T)) \\ = & \\ & F.R \blacktriangledown (F.S \vartriangle F.T)) \ . \end{array}$$

By equating the dummies R and S it follows that

$$zip_{F,\ominus} \ = \ I+(F.\ll \vartriangle F.\gg) \circ (F.\hookrightarrow \triangledown F.\hookleftarrow)^{\cup}$$

In particular, taking F to be product (see exercise 65) we obtain

$$zip_{\times,PL} \ = \ [\![I+((\ll \times \ll) \vartriangle (\gg \times \gg)) \circ ((\hookrightarrow \times \hookrightarrow) \triangledown (\hookleftarrow \times \hookleftarrow))^{\cup}]\!] \ .$$

7.5 Summary

For later use we summarise the zips so far constructed.

$$\begin{array}{lrcl} (80) & zip_{F,+} & = & F.\hookrightarrow^{\cup} \blacktriangledown F.\hookleftarrow^{\cup} \ , \\ (81) & zip_{F,\times} & = & F.\ll \vartriangle F.\gg \ , \\ (82) & zip_{F,G\bullet H} & = & G.zip_{F,H} \circ zip_{F,G} \ , \\ (83) & zip_{F,\dagger} & = & [\![I\ominus;\ zip_{F,\ominus}]\!] \ . \end{array}$$

(Note that (82) is the solution to exercise 77.)

8 Vetting the Candidates

As forewarned, three of the clauses in the specification of zips were postponed until after the construction of suitable candidates. In this section we verify that they are indeed satisfied in each case. Many verifications are entirely straightforward and we shall omit several parts.

8.1 Zips are Imps

The only element of the verification that all candidate zips are imps is, surprisingly, the verification that the zip constructed for disjoint sum is an imp. It is during this calculation that the restriction to bottom-strict relators F emerges.

The problem arises because $\blacktriangledown$ does not necessarily preserve imps. Formally, we prove:

$$F.\perp\!\!\!\perp \;=\; \perp\!\!\!\perp \quad\equiv\quad zip_{F,+} \text{ is an imp } .$$

The principal rule we use is

$$imp.(R \blacktriangledown S) \;\;\equiv\;\; imp.R \;\wedge\; imp.S \;\wedge\; R \circ S^{\cup} \;=\; \perp\!\!\!\perp \;\; .$$

Since $zip_{F,+} \;\;=\;\; F{\hookrightarrow^{\cup}} \blacktriangledown F.{\hookleftarrow^{\cup}}$ and both ${\hookrightarrow^{\cup}}$ and ${\hookleftarrow^{\cup}}$ are bijections (and thus imps) it suffices to show that $F.{\hookrightarrow^{\cup}} \circ (F.{\hookleftarrow^{\cup}})^{\cup} \;\;=\;\; \perp\!\!\!\perp$. But

$$
\begin{array}{ll}
 & F.{\hookrightarrow^{\cup}} \;\circ\; (F.{\hookleftarrow^{\cup}})^{\cup} \\
= & \qquad \{ \;\; \text{reverse} \;\; \} \\
 & F.{\hookrightarrow^{\cup}} \;\circ\; F.{\hookleftarrow} \\
= & \qquad \{ \;\; \text{relator}.F \;\; \} \\
 & F.({\hookrightarrow^{\cup}} \;\circ\; {\hookleftarrow}) \\
= & \qquad \{ \;\; \text{injections} \;\; \} \\
 & F.\perp\!\!\!\perp \;\; .
\end{array}
$$

Thus, $zip_{F,+}$ is an imp if and only if F is bottom-strict.

8.2 Zips are Natural Transformations

In this subsection we verify (14) for each relator G in the class ζ excluding composition of relators. In detail the statements that we have to prove are as follows:

(84) $\quad F.{\hookrightarrow^{\cup}} \blacktriangledown F.{\hookleftarrow^{\cup}} \;\circ\; F.(R+S) \;\;=\;\; F.R+F.S \;\circ\; F.{\hookrightarrow^{\cup}} \blacktriangledown F.{\hookleftarrow^{\cup}} \;\;,$

$$(85)\qquad F.\ll\triangle F.\gg\ \circ\ F.(R\times S)\ \ =\ \ F.R\times F.S\ \circ\ F.\ll\triangle F.\gg\ \ ,$$

$$(86)\qquad [\![I\ominus;\ zip_{F,\ominus}]\!]\ \circ\ F.\dagger.R\ \ =\ \ \dagger.F.R\ \circ\ [\![I\ominus;\ zip_{F,\ominus}]\!]\ \ .$$

In each case F is a $\sqcap$-junctive relator and the identities are universally quantified over the free variables R and S. In (86) $\dagger$ is the relator given by $\dagger.R\ =\ \mu(R\ominus)$. The proofs of (84) and (85) follow from exercise 66.

The proof of (86) we begin by making the inductive hypothesis:

$$(87)\qquad \forall(R,S :: zip_{F,\ominus}\ \circ\ F.(R\ominus S)\ \ =\ \ F.R\ominus F.S\ \circ\ zip_{F,\ominus})\ \ .$$

Then we have:

$$
\begin{array}{ll}
 & [\![I\ominus;\ zip_{F,\ominus}]\!]\ \circ\ F.\dagger.R\ \ =\ \ \dagger.F.R\ \circ\ [\![I\ominus;\ zip_{F,\ominus}]\!] \\
\equiv & \qquad \{\ \text{map fusion}\ \} \\
 & [\![I\ominus;\ zip_{F,\ominus}]\!]\ \circ\ F.\dagger.R\ \ =\ \ [\![I\ominus;\ F.R\ominus I\ \circ\ zip_{F,\ominus}]\!] \\
\Leftarrow & \qquad \{\ \text{anamorphism fusion}\ \} \\
 & zip_{F,\ominus}\ \circ\ F.\dagger.R\ \ =\ \ I\ominus F.\dagger.R\ \circ\ F.R\ominus I\ \circ\ zip_{F,\ominus} \\
\equiv & \qquad \{\ \dagger.S\ \ =\ \ S\ominus\dagger.S,\ \text{relators}\ \} \\
 & zip_{F,\ominus}\ \circ\ F.(R\ominus\dagger.R)\ \ =\ \ F.R\ominus F.\dagger.R\ \circ\ zip_{F,\ominus} \\
\Leftarrow & \qquad \{\ R,S := R,\dagger.R\ \} \\
 & \forall(R,S :: zip_{F,\ominus}\ \circ\ F.(R\ominus S)\ \ =\ \ F.R\ominus F.S\ \circ\ zip_{F,\ominus}) \\
\equiv & \qquad \{\ \ \text{induction hypothesis: (87)}\ \} \\
 & \text{true}\ \ .
\end{array}
$$

8.3 Zips have the Correct Domains

In this subsection we verify (15) for each relator G in the class of ζ , i.e. we have to verify:

$$(88)\qquad zip_{F,G}\ \circ\ F.G.I\ \ =\ \ zip_{F,G}\ \ =\ \ G.F.I\ \circ\ zip_{F,G}$$

Note that in the presence of (14):

$$(89)\qquad zip_{F,G}\ \circ\ F.G.I\ \ =\ \ zip_{F,G}\ \ \equiv\ \ zip_{F,G}\ \ =\ \ G.F.I\ \circ\ zip_{F,G}$$

So, in order to verify (15) it is sufficient to prove only one of the sides of the equivalence in (89).

In detail the statements that we have to prove are as follows:

$$(90)\qquad F.I+F.I\ \circ\ F.\hookrightarrow_{\cup}\blacktriangledown F.\hookleftarrow_{\cup}\ \ =\ \ F.\hookrightarrow_{\cup}\blacktriangledown F.\hookleftarrow_{\cup}$$

(91) $\quad F.I \times F.I \circ F.\ll \triangle F.\gg \quad = \quad F.\ll \triangle F.\gg$

(92) $\quad G.F.I \circ G.zip_{F,H} \circ zip_{F,G} \quad = \quad G.zip_{F,H} \circ zip_{F,G}$

(93) $\quad \dagger.F.I \circ [\![I\ominus;\ zip_{F,\ominus}]\!] \quad = \quad [\![I\ominus;\ zip_{F,\ominus}]\!]$

The proofs of (90) and (91) are trivial; just use generator fusion. The verification of (92) is equally trivial; use (15) for $zip_{F,H}$. Finally, (93) is verified by

$$
\begin{array}{ll}
 & \dagger.F.I \circ [\![I\ominus;\ zip_{F,\ominus}]\!] \\
= & \qquad \{ \text{ map fusion } \} \\
 & [\![I\ominus;\ F.I \ominus I \circ zip_{F,\ominus}]\!] \\
= & \qquad \{ \ zip_{F,\ominus} \ = \ F.I \ominus F.I \circ zip_{F,\ominus} \ \} \\
 & [\![I\ominus;\ F.I \ominus F.I \circ zip_{F,\ominus}]\!] \\
= & \qquad \{ \ F.I \ominus F.I \circ zip_{F,\ominus} \ = \ zip_{F,\ominus} \ \} \\
 & [\![I\ominus;\ zip_{F,\ominus}]\!] \ .
\end{array}
$$

9 Zip-a-dee-doo-dah, Zip-a-dee-ay

We are now ready for the final part of the proof of the zip-a-dee-doo-dah theorem stated in section 5, namely that pairs of relators in the class ζ are commuting. We will however omit consideration of clause (b) in the definition of ζ (see 40) but, however, see exercise 113. Our apologies for this shortcoming. (There are other inadequacies in the proof which will be pointed out later.)

Remember that all relators in ζ are $\sqcap$-junctive and bottom-strict. The preceding sections have therefore established the first part of the zip-a-dee-doo-dah theorem and it remains to show that if both relators F and G are in the class ζ then the constructed zips $zip_{F,G}$ and $zip_{G,F}$ are inverses of each other. This we also do by induction on the structure of ζ.

This leaves the verification of (18) for all ten combinations of relators F and G chosen from the four different types in the class ζ. Of these ten, two have been considered: in exercise 65 you were asked to show that $zip_{\times,\times} = (zip_{\times,\times})^{\cup}$. and $zip_{+,+} = (zip_{+,+})^{\cup}$. Forgetting about compositions of relators this leaves us with the obligation to prove the four identities:

(94) $\quad zip_{+,\times} \quad = \quad (zip_{\times,+})^{\cup}$,

$$(95) \qquad zip_{\dagger,+} \;=\; (zip_{+,\dagger})^{\cup} \;,$$

$$(96) \qquad zip_{\dagger,\times} \;=\; (zip_{\times,\dagger})^{\cup} \;,$$

$$(97) \qquad zip_{\dagger,\ddagger} \;=\; (zip_{\ddagger,\dagger})^{\cup} \;.$$

where $\dagger$ and $\ddagger$ are both map relators: $\dagger.R \;=\; \mu(R\ominus)$ and $\ddagger.R \;=\; \mu(R\oplus)$.

These are to be proved under the inductive hypothesis that the zip-a-dee-doo-dah theorem is valid for the following combinations of F and G: in the case of (95), $F = +$ and $G = \ominus$, in the case of (96), $F = \times$ and $G = \ominus$, and in the case of (97), $F = \dagger$ and $G = \oplus$, $F = \ddagger$ and $G = \ominus$, and $F = \ominus$ and $G = \oplus$.

We take responsibility for the proof of (94) and (97), the remaining proofs are delegated to the reader.

Exercise 98 Prove properties (95) and (96). (Hint: try to prove them both at once using exercise 66, i.e. prove that

$$[\![I\ominus;\; zip_{F,\ominus}]\!]^{\cup} \circ \dagger.\hat{F}.R \;=\; \hat{F}.\dagger.R$$

and use (93).

□

9.1 Product and Sum

First let us consider (94). We have to prove that

$$(99) \qquad (\ll + \ll) \vartriangle (\gg + \gg) \;=\; ((\hookrightarrow^{\cup} \times \hookrightarrow^{\cup}) \blacktriangledown (\hookleftarrow^{\cup} \times \hookleftarrow^{\cup}))^{\cup}$$

By the definition of co-junc (99) reduces to

$$(100) \qquad (\ll + \ll) \vartriangle (\gg + \gg) \;=\; (\hookrightarrow \times \hookrightarrow) \triangledown (\hookleftarrow \times \hookleftarrow)$$

which we verify as follows:

$$\begin{array}{ll} & (\ll + \ll) \vartriangle (\gg + \gg) \;=\; (\hookrightarrow \times \hookrightarrow) \triangledown (\hookleftarrow \times \hookleftarrow) \\ \equiv & \qquad \{ \text{ junc elimination } \} \\ & (\ll + \ll) \vartriangle (\gg + \gg) \circ \hookleftarrow \;=\; \hookleftarrow \times \hookleftarrow \\ \wedge & (\ll + \ll) \vartriangle (\gg + \gg) \circ \hookrightarrow \;=\; \hookrightarrow \times \hookrightarrow \quad . \end{array}$$

Continuing with just the first conjunct:

$$
\begin{array}{cl}
 & (\ll + \ll) \vartriangle (\gg + \gg) \circ \hookleftarrow \\
= & \qquad \{ \ \hookleftarrow \text{ is an imp, split-imp fusion } \} \\
 & (\ll + \ll \circ \hookleftarrow) \vartriangle (\gg + \gg \circ \hookleftarrow) \\
= & \qquad \{ \ \text{computation rule } \} \\
 & (\hookleftarrow \circ \ll) \vartriangle (\hookleftarrow \circ \gg) \\
= & \qquad \{ \ \text{definition } \} \\
 & \hookleftarrow \times \hookleftarrow
\end{array}
$$

The other conjunct is proved similarly so the two zips are indeed reverses of each other.

9.2 Maps

We have to prove that

$$(101) \quad [\![I\oplus;\ zip_{\dagger,\oplus}]\!] \quad = \quad [\![I\ominus;\ zip_{\ddagger,\ominus}]\!]^{\cup}.$$

We begin by reducing (101) to a more symmetrical formula. We have

$$
\begin{array}{cl}
 & [\![I\oplus;\ zip_{\dagger,\oplus}]\!] \\
= & \qquad \{ \ \text{induction hypothesis applied to } F = \dagger \text{ and } G = \oplus \ \} \\
 & [\![I\oplus;\ (zip_{\oplus,\dagger})^{\cup}]\!] \\
= & \qquad \{ \ zip_{\oplus,\dagger} \ = \ [\![I\ominus;\ zip_{\oplus,\ominus}]\!], \text{ definition of anamorphism } \} \\
 & [\![I\oplus;\ (\![I\ominus;\ (zip_{\oplus,\ominus})^{\cup}]\!)]\!] \quad .
\end{array}
$$

By interchanging $\dagger$ and $\ddagger$, and $\ominus$ and $\oplus$, we also have

$$[\![I\ominus;\ zip_{\ddagger,\ominus}]\!] \quad = \quad [\![I\ominus;\ (\![I\oplus;\ (zip_{\ominus,\oplus})^{\cup}]\!)]\!] \quad .$$

Hence,

$$
\begin{array}{cl}
 & (97) \\
\equiv & \qquad \{ \ \text{above } \} \\
 & [\![I\oplus;\ (\![I\ominus;\ (zip_{\oplus,\ominus})^{\cup}]\!)]\!] \quad = \quad [\![I\ominus;\ (\![I\oplus;\ (zip_{\ominus,\oplus})^{\cup}]\!)]\!]^{\cup} \\
\equiv & \qquad \{ \ \text{lhs: by induction hypothesis } \ (zip_{\oplus,\ominus})^{\cup} \ = \ zip_{\ominus,\oplus} \\
 & \qquad \quad \text{rhs: definition of anamorphism } \} \\
 & [\![I\oplus;\ (\![I\ominus;\ zip_{\ominus,\oplus}]\!)]\!] \quad = \quad (\![I\ominus;\ [\![I\oplus;\ zip_{\ominus,\oplus}]\!]]\!) \quad .
\end{array}
$$

We have thus reduced the proof obligation to one of showing that

$$(102)\qquad [\![I\oplus;\ (\!| I\ominus;\ zip_{\ominus,\oplus} |\!)]\!] \quad = \quad [\![I\ominus;\ (\!| I\oplus;\ zip_{\ominus,\oplus} |\!)]\!] \quad .$$

In order to prove (102) we abstract from its specific details. We note that it takes the form

$$(103)\qquad [\![F;\ (\!| G;\ \gamma_1 |\!)]\!] \quad = \quad (\!| G;\ [\![F;\ \gamma_2]\!] |\!)$$

for relators F and G and specs γ_1 and γ_2. This raises the question of what condition or conditions on the pair of specs γ_1, γ_2 are sufficient to guarantee that the equality holds. (Note that in (102) γ_1 and γ_2 are the same. We use subscripts here rather than, say, α and β so that you may turn a blind eye to them if you wish. The extra generality in the theorem we are about to prove is not needed in this application.) Investigation of this question results in the following theorem.

Theorem 104 (Fokkinga's Theorem)

$$\begin{array}{ll} & [\![F;\ (\!| G;\ \gamma_1 |\!)]\!] \quad = \quad (\!| G;\ [\![F;\ \gamma_2]\!] |\!) \\ \Leftarrow & \forall (R :: \ F.G.R \circ \gamma_1 \quad = \quad \gamma_2 \circ G.F.R) \quad . \end{array}$$

□

We attribute this theorem to Fokkinga although he didn't formulate or prove the theorem at this level of generality — he proved a particular case in which F was instantiated to stream and G to list. Nevertheless, it was clear to us when reading his calculation that essentially the same calculation could be carried out at a more general level (with the added advantage that the calculation becomes much crisper).

Proof of Fokkinga's theorem obviously entails the use of the unique extension property. But we have a dilemma here. Do we apply the uep to the F-anamorphism on the left side of the consequent or to the G-catamorphism on the right side of the consequent? Following Fokkinga's lead we formulate a lemma that obviates making a choice.

Lemma 105 (Fokkinga's Lemma) Let p and q be predicates on specs, and f and g be functions from specs to specs. Suppose the equations

$$(106)\qquad X :: \qquad X = f.X \quad \wedge \quad p.X$$

(107) $X ::\quad X = g.X \;\wedge\; q.X$

both have unique solutions denoted by $\bar{f}$ and $\bar{g}$, respectively. Then

(108) $\bar{f} = \bar{g} \;\Leftarrow\; \forall(X ::\; f.g.X \;=\; g.f.X) \wedge p.\bar{f} \wedge q.f.\bar{g}$

Note: Essentially this is Fokkinga's theorem 13 [1, p. 89]. In introducing predicates p and q one rectifies an omission in his subsequent application of the theorem. Specifically, he fails to observe later the context in which he demands uniqueness (within an F-algebra or within a G-coalgebra). *End of note.*

Proof

$$
\begin{array}{ll}
 & \bar{f} \;=\; \bar{g} \\
\equiv & \qquad \{\; \bar{f} \text{ is unique soln. of (106) } \} \\
 & \bar{g} \;=\; f.\bar{g} \;\;\wedge\;\; p.\bar{g} \\
\equiv & \qquad \{\; \bar{g} \text{ is unique soln. of (107) } \} \\
 & f.\bar{g} \;=\; g.f.\bar{g} \;\;\wedge\;\; q.f.\bar{g} \;\;\wedge\;\; p.\bar{g} \\
\equiv & \qquad \{\; \forall(X ::\; f.g.X \;=\; g.f.X) \;\} \\
 & f.\bar{g} \;=\; f.g.\bar{g} \;\;\wedge\;\; q.f.\bar{g} \;\;\wedge\;\; p.\bar{g} \\
\equiv & \qquad \{\; \bar{g} \text{ solves (107), Leibniz } \} \\
 & q.f.\bar{g} \;\;\wedge\;\; p.\bar{g}
\end{array}
$$

□

We apply the above lemma to the proof of Fokkinga's theorem in two steps. First suppose α and β are arbitrary specs and define

$$
\begin{array}{lll}
f.X & = & F.X \circ \alpha \\
g.X & = & \beta \circ G.X \;\; .
\end{array}
$$

Also let $p.X \;\equiv\; X = \mu F \circ X$ and $q.X \;\equiv\; X = X \circ \mu G$. With these definition (106) and (107) in Fokkinga's lemma have unique solutions, namely $[\![F;\ \alpha]\!]$ and $(\!|G;\ \beta|\!)$ respectively. So by application of the lemma we have

$$[\![F;\ \alpha]\!] \;=\; (\!|G;\ \beta|\!) \;\;\Leftarrow\;\; (109) \;\wedge\; (110) \;\wedge\; (111)$$

where

$$
\begin{array}{lll}
(109) & \equiv & \forall(X ::\; f.g.X \;=\; g.f.X) \;\; , \\
(110) & \equiv & (\!|G;\ \beta|\!) \;\;=\;\; \mu F \circ \;\; (\!|G;\ \beta|\!) \\
(111) & \equiv & F.(\!|G;\ \beta|\!) \circ \alpha \;\;=\;\; F.(\!|G;\ \beta|\!) \circ \alpha \circ \mu G \;\; .
\end{array}
$$

This implication remains valid if we strengthen its antecedent. Property (110) can be strengthened by

$$\begin{array}{ll}
 & (110) \\
\equiv & \quad \{ \;\text{catamorphism computation rule}\; \} \\
 & \beta \circ G.(\!|G;\ \beta|\!) \;=\; \mu F \circ \beta \circ G.(\!|G;\ \beta|\!) \\
\Leftarrow & \quad \{ \;\text{Leibniz}\; \} \\
 & \beta \;=\; \mu F \circ \beta
\end{array}$$

Also, by Leibniz's rule,

$$(111) \quad \Leftarrow \quad \alpha = \alpha \circ \mu G \ .$$

The conclusion of this first step is thus

$$\begin{array}{lll}
(112) & & [\![F;\ \alpha]\!] \;=\; (\!|G;\ \beta|\!) \\
 & \Leftarrow & (109) \;\wedge\; \beta = \mu F \circ \beta \;\wedge\; \alpha = \alpha \circ \mu G \ .
\end{array}$$

The next step is to instantiate β to $[\![F;\ \gamma_2]\!]$ and α to $(\!|G;\ \gamma_1|\!)$. By the domain rule for catamorphisms and its dual for anamorphisms this choice discharges the second and third conjuncts in the antecedent of (112). It remains to simplify (109).

$$\begin{array}{ll}
 & (109) \\
\equiv & \quad \{ \;\text{definitions of } f \text{ and } g\; \} \\
 & \forall(X :: \; F.(\beta \circ G.X) \circ \alpha \;=\; \beta \circ G.(F.X \circ \alpha)) \\
\equiv & \quad \{ \;\text{definitions of } \alpha \text{ and } \beta\text{, relators}\; \} \\
 & \forall(X :: \; F.[\![F;\ \gamma_2]\!] \circ F.G.X \circ (\!|G;\ \gamma_1|\!) \\
 & \qquad\qquad = \;[\![F;\ \gamma_2]\!] \circ G.F.X \circ G.(\!|G;\ \gamma_2|\!))
\end{array}$$

But, for all specs X,

$$\begin{array}{ll}
 & F.[\![F;\ \gamma_2]\!] \circ F.G.X \circ (\!|G;\ \gamma_1|\!) \\
 & \quad = \;[\![F;\ \gamma_2]\!] \circ G.F.X \circ G.(\!|G;\ \gamma_1|\!) \\
\equiv & \quad \{ \;\text{computation rules for cata and anamorphisms}\; \} \\
 & F.[\![F;\ \gamma_2]\!] \circ F.G.X \circ \gamma_1 \circ G.(\!|G;\ \gamma_1|\!) \\
 & \quad = \;F.[\![F;\ \gamma_2]\!] \circ \gamma_2 \circ G.F.X \circ G.(\!|G;\ \gamma_1|\!) \\
\Leftarrow & \quad \{ \;\text{Leibniz}\; \} \\
 & F.G.X \circ \gamma_1 \;=\; \gamma_2 \circ G.F.X \ .
\end{array}$$

This concludes the proof of Fokkinga's theorem.

After this longish interlude we can return to the proof of (102).

$$
\begin{array}{ll}
 & [\![I\oplus;\ (\!| I\ominus;\ zip_{\ominus,\oplus} |\!)]\!] \quad = \quad (\!| I\ominus;\ [\![I\oplus;\ zip_{\ominus,\oplus}]\!] |\!) \\
\Leftarrow & \qquad \{ \text{ Fokkinga's theorem } \} \\
 & \forall(R ::\ I \oplus (I \ominus R) \circ zip_{\ominus,\oplus} \quad = \quad zip_{\ominus,\oplus} \circ I \ominus (I \oplus R)) \\
\equiv & \qquad \{ \text{ by the induction hypothesis} \\
 & \qquad\quad zip_{\ominus,\oplus} \quad = \quad (I \ominus I) \oplus (I \ominus I) \circ zip_{\ominus,\oplus} \\
 & \qquad\quad \text{and } zip_{\ominus,\oplus} \quad = \quad zip_{\ominus,\oplus} \circ (I \oplus I) \ominus (I \oplus I) \quad . \\
 & \qquad\quad \text{See the discussion of (15) and (18)} \quad \} \\
 & \forall(R ::\ (I \ominus I) \oplus (I \ominus R) \circ zip_{\ominus,\oplus} \\
 & \qquad\qquad = \quad zip_{\ominus,\oplus} \circ (I \oplus I) \ominus (I \oplus R)) \\
\equiv & \qquad \{ \text{ by the induction hypothesis} \\
 & \qquad\quad (S \ominus T) \oplus (U \ominus V) \circ zip_{\ominus,\oplus} \\
 & \qquad\qquad = \quad zip_{\ominus,\oplus} \circ (S \oplus U) \ominus (T \oplus V), \\
 & \qquad\quad S,T,U,V := I,I,I,R \quad \} \\
 & true \quad .
\end{array}
$$

And that completes the proof of the zip-a-dee-doo-dah theorem!

Exercise 113 A major shortcoming of the above proof is the absence of a formalisation of the composition of relators and proofs of the relevant lemmas. A simple instance of composition is when F and G are both unary and we form $F \bullet G$ (defined, of course, by $(F \bullet G).R \;=\; F.(G.R)$). This exercise asks you to prove the lemmas relevant to this form of relator compostion. (Note, however, that such a composition does not enable the construction of, for example, the ternary relator $R,S,R \mapsto R + (S \times T)$ from the binary relators $+$ and $\times$.)

The answer to exercise 77 given earlier is needed for this question:

(a) $$zip_{F,G\bullet H} \quad = \quad G.zip_{F,H} \circ zip_{F,G} \quad .$$

The exercise is to verify the zip-a-dee-doo-dah theorem with G instantiated to $G \bullet H$. In particular verify that

(b) $$zip_{F,G\bullet H} \circ F.G.H.R \quad = \quad G.H.F.R \circ zip_{F,G\bullet H}$$

(c) $$zip_{F,G\bullet H} \circ F.G.H.I \quad = \quad zip_{F,G\bullet H}$$

assuming that

(d) $\quad zip_{F,G} \circ F.G.R \;=\; G.F.R \circ zip_{F,G}$
(e) $\quad zip_{F,G} \circ F.G.I \;=\; zip_{F,G}$
(f) $\quad zip_{F,H} \circ F.H.R \;=\; H.F.R \circ zip_{F,H}$
(g) $\quad zip_{F,H} \circ F.H.I \;=\; zip_{F,H}$

(That $zip_{F,G\bullet H}$ is an imp is trivial.)

Verify also that

(h) $\quad zip_{+,G\bullet H} \;=\; (zip_{G\bullet H,+})^{\cup}$
(i) $\quad zip_{\times,G\bullet H} \;=\; (zip_{G\bullet H,\times})^{\cup}$
(j) $\quad zip_{\dagger,G\bullet H} \;=\; (zip_{G\bullet H,\dagger})^{\cup}$

(where $\dagger$ is the map relator $\dagger.R \;=\; \mu(R\ominus)$).
In each case make the appropriate inductive hypothesis.
(Hint: for (h) and (i) use exercise 66 as you did in exercise 98.)

□

Exercise 114 Excluded from the class ζ are all those relators constructed using the unit type $\mathbb{1}$ because, it was argued, it is not possible to "unzip" an empty structure. Unfortunately this excludes some relators defined in terms of the unit type but which nevertheless can never generate an empty structure. For example, consider the relator *Rose* defined such that $Rose.R$ is the least solution of the equation

$$X :: \qquad X \;=\; R \times List.X$$

where $List.R$ is the least solution of

$$X :: \qquad X \;=\; \mathbb{1} + R \times X$$

Roses are never empty but they are excluded from the relators shown to be zippable in this paper because *Rose* is defined in terms of *List* which, in turn, is defined in terms of the unit type.

The way out of this problem is to observe that *List* is naturally isomorphic to the relator $(X \;\mapsto\; \mathbb{1}{+}PL.X)$, PL being the relator defining positive lists, and hence (since $\times$ distributes over $+$ and $\mathbb{1}$ is a unit of $\times$, up to natural isomorphism) the relator *Rose* is naturally isomorphic to the relator *Roos* defined such that $Roos.R$ is the least solution of the equation

$$X :: \qquad X \;=\; R \,+\, R \times PL.R \quad .$$

Note that *Roos* is defined in terms of relators that are known to be zippable.

To bridge the gap a theorem is needed establishing that the class of zippable relators is closed under natural isomorphisms.

Suppose that γ witnesses a natural isomorphism between F and G. Suppose also that G has generator $\hat{G}$. Show how to construct the generator $\hat{F}$ satisfying (20), (21) and (22). Show also how to construct $zip_{H,F}$ for all relators H for which it is known that $zip_{H,G}$ exists.

□

Open Problem

As we know, $zip_{F,\times} \;=\; F.\ll \vartriangle F.\gg$ satisfies

(115) $\quad zip_{F,\times} \circ F.(R \vartriangle S) \;=\; F.R \vartriangle F.S$

and

(116) $\quad zip_{F,\times} \circ F.(I \times I) \;=\; zip_{F,\times} \quad .$

The converse is also true. Any spec α satisfying

(117) $\quad \alpha \circ F.(R \vartriangle S) \;=\; F.R \vartriangle F.S$

and

(118) $\quad \alpha \circ F.(I \times I) \;=\; \alpha$

must be equal $zip_{F,\times}$ since

$$\begin{array}{ll}
 & (117) \\
\Leftarrow & \quad \{ \;\; R,S := \;\ll,\gg \;\; \} \\
 & \alpha \circ F.(\ll \vartriangle \gg) \;=\; F.\ll \vartriangle F.\gg \\
\equiv & \quad \{ \;\; \ll \vartriangle \gg \;=\; I \times I, (118) \;\; \} \\
 & \alpha \;=\; F.\ll \vartriangle F.\gg
\end{array}$$

In other words $zip_{F,\times}$ is uniquely defined by (115) and (116). The open question is: is $zip_{F,G}$ uniquely determined by the requirements (13) – (16).

□

10 Banana Split and Other Delights

The existence of *zips* is surprisingly "fruitful". In this final section we mention two beautiful concrete consequences of the existence of $zip_{F,+}$ and $zip_{F,\times}$ for all $\sqcap$-junctive F. We also hint at other consequences of their existence in the exercises. We do not list all consequences that we know of, and we have good reason to believe that there are lots more that we do not know of at this point in time!

One delightful and surprising reward of this equality is that $\vartriangle$ and $\triangledown$ abide with each other.

$$(119)\quad (R \triangledown S) \vartriangle (T \triangledown U) \quad = \quad (R \vartriangle T) \triangledown (S \vartriangle U)$$

since

$$\begin{array}{cl}
 & (R \triangledown S) \vartriangle (T \triangledown U) \\
= & \qquad \{\ \times - \vartriangle \text{ fusion and } \triangledown - + \text{ fusion }\ \} \\
 & (R \triangledown S) \times (T \triangledown U) \ \circ\ (I+I) \vartriangle (I+I) \\
= & \qquad \{\ (34)\ \} \\
 & (R \triangledown S) \times (T \triangledown U) \ \circ\ zip_{+,\times} \ \circ\ (I \vartriangle I) + (I \vartriangle I) \\
= & \qquad \{\ zip_{+,\times} \ = \ (zip_{\times,+})^{\cup}\ \} \\
 & (R \triangledown S) \times (T \triangledown U) \ \circ\ (zip_{\times,+})^{\cup} \ \circ\ (I \vartriangle I) + (I \vartriangle I) \\
= & \qquad \{\ (34)\ \} \\
 & (R \times T) \triangledown (S \times U) \ \circ\ (I \vartriangle I) + (I \vartriangle I) \\
= & \qquad \{\ \triangledown - + \text{ fusion and } \times - \vartriangle \text{ fusion }\ \} \\
 & (R \vartriangle T) \triangledown (S \vartriangle U)
\end{array}$$

Another reward is the theorem affectionately known as the "banana-split rule" [1].

$$(120)\quad (\!|F;\ R|\!) \vartriangle (\!|F;\ S|\!) \quad = \quad (\!|F;\ R \times S \ \circ\ F.\!\ll \vartriangle F.\!\gg|\!)$$

for all $\sqcap$-junctive relators F.
Verification of (120) involves the use of the unique extension property for F-catamorphisms. According to the uep we have to establish two facts

$$(121)\quad (\!|F;\ R|\!) \vartriangle (\!|F;\ S|\!) \quad = \quad (\!|F;\ R|\!) \vartriangle (\!|F;\ S|\!) \ \circ\ \mu F \quad ,$$

and

$$(122)\quad (\!|F;\ R|\!) \vartriangle (\!|F;\ S|\!) \quad = \quad R \times S \ \circ\ F.\!\ll \vartriangle F.\!\gg \ \circ\ F.((\!|F;\ R|\!) \vartriangle (\!|F;\ S|\!))$$

Equation (121) follows from split-imp fusion (μF is a monotype and thus an imp) and the domain rule for catamorphisms. For (122) we have

$$
\begin{array}{ll}
 & (\![F;\ R]\!) \vartriangle (\![F;\ S]\!) \\
= & \qquad \{ \quad \text{catamorphism computation rule} \quad \} \\
 & (R \circ F.(\![F;\ R]\!)) \vartriangle (S \circ F.(\![F;\ S]\!)) \\
= & \qquad \{ \ \times - \vartriangle \text{ fusion } \ \} \\
 & R \times S \circ F.(\![F;\ R]\!) \vartriangle F.(\![F;\ S]\!) \\
= & \qquad \{ \ F \text{ is } \sqcap\text{-junctive, (57)} \ \} \\
 & R \times S \circ (F.\!\ll \vartriangle F.\!\gg) \circ F.((\![F;\ R]\!) \vartriangle (\![F;\ S]\!)) \quad .
\end{array}
$$

Exercise 123 Both the $\triangledown$-$\vartriangle$ abide law and the banana split theorem are instances of more general properties predicted by the existence of zips. In this exercise these properties are stated for unary relators and you are asked to prove them. All you need to know are the properties stated here and the uep and domain law of catamorphisms.

Recall that the F-generator $\hat{F}$ has the property

(a) $\qquad \hat{F}.(R \circ S) \;=\; F.R \circ \hat{F}.S$

and, for all imps f,

(b) $\qquad \hat{F}.R \circ f \;=\; \hat{F}.(R \circ f)$.

We had as definition $\check{G}.R \;=\; (\hat{G}.(R\cup))\cup$ and

(c) $\qquad \check{G}.(R \circ S) \;=\; \check{G}.R \circ G.S$.

Suppose that $zip_{F,G}$ and $zip_{G,F}$ both exist and are reverses of each other. I.e. assume the existence of γ such that

(d) $\qquad F.\check{G}.R \circ \gamma \;=\; \check{G}.F.R$

and

(e) $\qquad \hat{F}.G.R \;=\; \gamma \circ G.\hat{F}.R$

Prove that

(f) $\qquad \hat{F}.\check{G}.R \;=\; \check{G}.\hat{F}.R$

(g) $\qquad \hat{F}.(\![G;\ R]\!) \;=\; (\![G;\ F.R \circ \gamma]\!)$.

Property (g) dualises to a property of $\check{F}$ and G-anamorphisms. What is that property?
□

Exercise 124 The banana split theorem has a dual for disjoint sum. Identify the theorem and prove it. (The restriction on F is not necessary.) (Hint: use exercise 123.)
□

Exercise 125 The existence of $zip_{+,+}$ and $zip_{\times,\times}$, each of which is equal to its own inverse, predicts two further abide laws. What are these laws? (Hint: use exercise 123.)
□

Exercise 126 We have the following distribution law: if T distributes over $P \vartriangle U$:

(a) $$P \vartriangle U \;\circ\; T \;=\; (P \circ T) \;\vartriangle\; (U \circ T)$$

and S distributes over $U \vartriangle Q$:

(b) $$U \vartriangle Q \;\circ\; S \;=\; (U \circ S) \;\vartriangle\; (Q \circ S)$$

then we also have:

(c) $$\begin{array}{ll} & (P \times Q) \;\vartriangle\; (U \times V) \;\circ\; T \times S \\ = & \\ & ((P \circ T) \times (Q \circ S)) \;\vartriangle\; ((U \circ T) \times (V \circ S)) \end{array}$$

and

(d) $$\begin{array}{ll} & (P \times Q) \;\vartriangle\; (U \times V) \;\circ\; T \vartriangle S \\ = & \\ & ((P \circ T) \;\vartriangle\; (Q \circ S)) \;\vartriangle\; ((U \circ T) \;\vartriangle\; (V \circ S)) \end{array}$$

Both follow from the properties of $zip_{\times,\times}$:

$$
\begin{array}{ll}
 & (P\times Q) \,\vartriangle\, (U\times V) \circ T\times S \\
= & \quad \{ \text{ definition } \mathit{zip}_{\times,\times} \text{ (34) } \} \\
 & \mathit{zip}_{\times,\times} \circ (P \vartriangle U)\times(Q \vartriangle V) \circ T\times S \\
= & \quad \{ \ \times\text{-fusion } \} \\
 & \mathit{zip}_{\times,\times} \circ (P \vartriangle U\circ T)\times(Q \vartriangle V\circ S) \\
= & \quad \{ \text{ assumption (a) and (b) } \} \\
 & \mathit{zip}_{\times,\times} \circ ((P\circ T) \vartriangle (U\circ T))\times((Q\circ S) \vartriangle (V\circ S)) \\
= & \quad \{ \text{ definition } \mathit{zip}_{\times,\times} \text{ (34) } \} \\
 & ((P\circ T)\times(Q\circ S)) \vartriangle ((U\circ T)\times(V\circ S))
\end{array}
$$

and (d) follows from (c):

$$
\begin{array}{ll}
 & (P\times Q) \vartriangle (U\times V) \circ T \vartriangle S \\
= & \quad \{ \ \times\text{-}\vartriangle \text{ fusion } \} \\
 & (P\times Q) \vartriangle (U\times V) \circ T\times S \circ I \vartriangle I \\
= & \quad \{ \text{ (a) } \} \\
 & ((P\circ T)\times(Q\circ S)) \vartriangle ((U\circ T)\times(V\circ S)) \circ I \vartriangle I \\
= & \quad \{ \ I \vartriangle I \text{ is an imp, split-imp fusion } \} \\
 & ((P\circ T)\times(Q\circ S) \circ I \vartriangle I) \vartriangle ((U\circ T)\times(V\circ S) \circ I \vartriangle I) \\
= & \quad \{ \ \times\text{-}\vartriangle \text{ fusion } \} \\
 & ((P\circ T) \vartriangle (Q\circ S)) \vartriangle ((U\circ T) \vartriangle (V\circ S))
\end{array}
$$

These two laws are again an instance of a more general property predicted by the existence of zips. The exercise is to formulate and to prove these two properties. Note that in general $\hat{G}.I$ is not an imp (in the case of split it is), so you get an extra condition for the general case of (d).
□

11 Acknowledgement

Jaap van der Woude and Ed Voermans were the first to prove (60) (in particular to identify the requirement that F be $\sqcap$-junctive).

12 Solutions to the Exercises

Solution to exercise: 4
Property (7) we prove by:

$$
\begin{array}{ll}
 & \top\!\!\top \vartriangle \top\!\!\top \\
= & \quad \{ \text{ definition } \vartriangle \ \} \\
 & \ll^{\cup} \circ \top\!\!\top \sqcap \gg^{\cup} \circ \top\!\!\top \\
= & \quad \{ \text{ domains: } {\gg}{>} \ = \ {\ll}{>} \ = \ I{\times}I \ \} \\
 & I{\times}I \circ \top\!\!\top \sqcap I{\times}I \circ \top\!\!\top \\
= & \quad \{ \text{ calculus, domains: } F.I \circ \top\!\!\top \ = \ F.\top\!\!\top \circ \top\!\!\top , \\
 & \quad\ \ F.R \ = \ R \times R \ \} \\
 & \top\!\!\top{\times}\top\!\!\top \circ \top\!\!\top \quad .
\end{array}
$$

Property (1) is proved in a similar way:

$$
\begin{array}{ll}
 & \top\!\!\top{\times}\top\!\!\top \\
= & \quad \{ \text{ definition } \times \ \} \\
 & \ll^{\cup} \circ \top\!\!\top \circ \ll \sqcap \gg^{\cup} \circ \top\!\!\top \circ \gg \\
= & \quad \{ \text{ domains: } {\gg}{>} \ = \ {\ll}{>} \ = \ I{\times}I \ \} \\
 & I{\times}I \circ \top\!\!\top \circ I{\times}I \\
= & \quad \{ \text{ domains: } F.I \circ \top\!\!\top \ = \ F.\top\!\!\top \circ \top\!\!\top , \ F.R \ = \ R \times R \ \} \\
 & \top\!\!\top{\times}\top\!\!\top \circ \top\!\!\top \circ \top\!\!\top{\times}\top\!\!\top \quad .
\end{array}
$$

□

Solution to exercise: 37
We only give the solution of (39):

$$
\begin{array}{ll}
 & R{\oplus}S \circ T\hat{\oplus}U \\
= & \quad \{ \ (25) \ \} \\
 & R{\oplus}S \circ T{\oplus}U \circ I\hat{\oplus}I \\
= & \quad \{ \text{ relator fusion } \} \\
 & (R \circ T){\oplus}(S \circ U) \circ I\hat{\oplus}I \\
= & \quad \{ \ (25) \ \} \\
 & (R \circ T)\hat{\oplus}(S \circ U) \quad .
\end{array}
$$

□

§6.1 Proof of the properties (26) and (27) of co-junc

Property (27) follows from spec-junc fusion. For property (26) we have:

$$\begin{array}{ll}
& \top\!\!\top \blacktriangledown \top\!\!\top \\
= & \quad \{ \text{ definition } \blacktriangledown \ \} \\
& \hookrightarrow \circ \top\!\!\top \sqcup \hookleftarrow \circ \top\!\!\top \\
= & \quad \{ \text{ domains } \} \\
& \top\!\!\top{+}\bot\!\!\bot \circ \top\!\!\top \sqcup \bot\!\!\bot{+}\top\!\!\top \circ \top\!\!\top \\
= & \quad \{ \circ \text{ distributes over } \sqcup \ \} \\
& (\top\!\!\top{+}\bot\!\!\bot \sqcup \bot\!\!\bot{+}\top\!\!\top) \circ \top\!\!\top \\
= & \quad \{ + \text{ is } \sqcup\text{-junctive } \} \\
& \top\!\!\top{+}\top\!\!\top \circ \top\!\!\top \quad .
\end{array}$$

□

Solution to exercise: 43

Properties (45) – (49) follow from the dual properties of catamorphisms. The last exercise we prove by:

$$\begin{array}{ll}
& (\!|F;\ R|\!) \circ [\![F;\ S]\!] \\
= & \quad \{ \text{ computation rule cata-, anamorphism } \} \\
& R \circ F.(\!|F;\ R|\!) \circ F.[\![F;\ S]\!] \circ S \\
= & \quad \{ \text{ relator fusion } \} \\
& R \circ F.((\!|F;\ R|\!) \circ [\![F;\ S]\!]) \circ S \quad .
\end{array}$$

□

Solution to exercise: 53

The generators of *Ash*, *Box*, *Elm*, and *Fir* are:

$$\hat{Ash}.R \quad = \quad [\![I\ominus;\ R \blacktriangledown (I \vartriangle R) \vartriangle I]\!]$$

where $R \ominus S \quad = \quad R{+}(S{\times}R){\times}S \quad .$

$$\hat{Box}.R \quad = \quad [\![I\ominus;\ \hat{P}L.I \vartriangle R]\!]$$

where $R \ominus S \quad = \quad PL.S{\times}R \quad .$

$$\hat{Elm}.R \quad = \quad \hat{P}L.\hat{Box}.R \quad .$$

$$\hat{Fir}.R \quad = \quad [\![I\ominus;\ R \blacktriangledown \hat{Ash}.I]\!]$$

where $R \ominus S \quad = \quad R{+}Ash.S \quad .$

□

Solution to exercise: 66

Property (69) we prove by:

$$\begin{array}{ll}
 & \gamma \\
= & \qquad \{\ (68)\ \} \\
 & \gamma \circ F.G.I \\
= & \qquad \{\ \text{assumption } G\ \} \\
 & \gamma \circ F.\hat{G}.\alpha \\
= & \qquad \{\ (67)\ \} \\
 & \hat{G}.F.\alpha \\
= & \qquad \{\ \text{definition } \check{G}\ (23)\ \} \\
 & (\check{G}.F.\alpha^{\cup})^{\cup} \quad .
\end{array}$$

For the naturality property (71) we have:

$$\begin{array}{ll}
 & G.F.R \ \circ \hat{G}.F.\alpha \\
= & \qquad \{\ \text{generator fusion}\ \} \\
 & \hat{G}.F.(R \circ \alpha) \\
= & \qquad \{\ (70)\ \} \\
 & \hat{G}.F.\alpha \circ F.\hat{G}.(R \circ \alpha) \\
= & \qquad \{\ \text{generator fusion}\ \} \\
 & \hat{G}.F.\alpha \circ F.G.R \circ F.\hat{G}.\alpha \\
= & \qquad \{\ \hat{G}.\alpha \ = \ G.I, \text{ remove } F.G.I\ \} \\
 & \hat{G}.F.\alpha \circ F.G.R \quad .
\end{array}$$

The domain property (72) is easy to verify:

$$\begin{array}{ll}
 & \hat{G}.F.\alpha \\
= & \qquad \{\ \text{generator fusion}\ \} \\
 & G.F.I \circ \hat{G}.F.\alpha \\
= & \qquad \{\ (71)\ \} \\
 & \hat{G}.F.\alpha \circ F.G.I \quad .
\end{array}$$

$\square$

Solution to exercise: 77

We have to construct $zip_{F,G \bullet H}$ such that

$$zip_{F,G \bullet H} \circ F.\hat{G}.\hat{H}.R \quad = \quad \hat{G}.\hat{H}.F.R$$

We begin wih the lefthand side and try to transform it to the right.

$$\begin{array}{ll} & zip_{F,G\bullet H} \circ F.\hat{G}.\hat{H}.R \\ = & \qquad \{ \ \bullet \ zip_{F,G\bullet H} \quad = \quad \alpha \circ zip_{F,G} \ , (13) \quad \} \\ & \alpha \circ \hat{G}.F.\hat{H}.R \\ = & \qquad \{ \ \bullet \ \alpha \quad = \quad G.\beta, \text{ generator fusion } \} \\ & \hat{G}.(\beta \circ F.\hat{H}.R) \\ = & \qquad \{ \ \bullet \ \beta \quad = \quad zip_{F,H} \ , (13) \ \} \\ & \hat{G}.\hat{H}.F.R \ . \end{array}$$

We have thus constructed

$$zip_{F,G \bullet H} \quad = \quad G.zip_{F,H} \circ zip_{F,G}$$

$\square$

Solution to exercise: 65

The formulae are, for $zip_{+,+}$,

$$(\hookrightarrow_\cup + \hookrightarrow_\cup) \ \triangledown \ (\hookleftarrow_\cup + \hookleftarrow_\cup)$$

and for $zip_{\times,\times}$

$$(\ll \times \ll) \ \vartriangle \ (\gg \times \gg)$$

We start with proving that $zip_{\times,\times}$ is its own inverse:

$$\begin{array}{ll} & (\ll\times\ll) \vartriangle (\gg\times\gg) \\ = & \qquad \{ \ \vartriangle \text{ is } \sqcap\text{-junctive } \} \\ & (\ll\times\top\!\!\top) \vartriangle (\top\!\!\top\times\gg) \ \sqcap \ (\top\!\!\top\times\ll) \vartriangle (\gg\times\top\!\!\top) \end{array}$$

We continue with the left conjunct:

$$
\begin{array}{ll}
 & (\ll \times \top\!\!\top) \vartriangle (\top\!\!\top \times \gg) \\
= & \qquad \{ \text{ definition } \vartriangle \ \} \\
 & \ll^{\cup} \circ \ll \times \top\!\!\top \sqcap \gg^{\cup} \circ \top\!\!\top \times \gg \\
= & \qquad \{ \ R \times \top\!\!\top \ = \ \ll^{\cup} \circ R \circ \ll, \ \top\!\!\top \times R \ = \ \gg^{\cup} \circ R \circ \gg \ \} \\
 & \ll^{\cup} \circ \ll^{\cup} \circ \ll \circ \ll \sqcap \gg^{\cup} \circ \gg^{\cup} \circ \gg \circ \gg \\
= & \qquad \{ \text{ converse } \} \\
 & (\ll^{\cup} \circ \ll^{\cup} \circ \ll \circ \ll \sqcap \gg^{\cup} \circ \gg^{\cup} \circ \gg \circ \gg)^{\cup} \\
= & \qquad \{ \text{ same steps back } \} \\
 & ((\ll \times \top\!\!\top) \vartriangle (\top\!\!\top \times \gg))^{\cup}
\end{array}
$$

The other conjunct is proved to be its own inverse in a similarly way:

$$
\begin{array}{ll}
 & (\top\!\!\top \times \ll) \vartriangle (\gg \times \top\!\!\top) \\
= & \qquad \{ \text{ definition } \vartriangle \ \} \\
 & \ll^{\cup} \circ \top\!\!\top \times \ll \sqcap \gg^{\cup} \circ \gg \times \top\!\!\top \\
= & \qquad \{ \ \top\!\!\top \times R \ = \ \gg^{\cup} \circ R \circ \gg, R \times \top\!\!\top \ = \ \ll^{\cup} \circ R \circ \ll \ \} \\
 & \ll^{\cup} \circ \gg^{\cup} \circ \ll \circ \gg \sqcap \gg^{\cup} \circ \ll^{\cup} \circ \gg \circ \ll \\
= & \qquad \{ \text{ converse } \} \\
 & (\ll^{\cup} \circ \gg^{\cup} \circ \ll \circ \gg \sqcap \gg^{\cup} \circ \ll^{\cup} \circ \gg \circ \ll)^{\cup} \\
= & \qquad \{ \text{ same steps back } \} \\
 & ((\top\!\!\top \times \ll) \vartriangle (\gg \times \top\!\!\top))^{\cup}
\end{array}
$$

From this we conclude that $zip_{\times,\times}$ is its own inverse. The proof that $zip_{+,+}$ is its own inverse is the dual of this proof.
□

Solution to exercise: 98
From exercise 66 we know that there exists only one solution for equations (67) and (68), i.e. taking $F := \dagger$ and $G := +$ (and adjusting for the fact that $+$ is a binary relator) we get the equations

(127) $\quad \gamma \circ \dagger.(R \blacktriangledown S) \quad = \quad \dagger.R \blacktriangledown \dagger.S$

(128) $\quad \gamma \circ I{+}I \quad = \quad \gamma$

which we know is solved by $zip_{\dagger,+}$. So, if we can prove that $(zip_{+,\dagger})^{\cup}$ also solves equations (127) and (128), we conclude that they are equal. The same holds for $zip_{\dagger,\times}$ and $(zip_{\times,\dagger})^{\cup}$. So, in general we have to prove:

$$[\![I\ominus;\ zip_{F,\ominus}]\!]^{\cup} \circ \dagger.\hat{F}.R \quad = \quad \hat{F}.\dagger.R$$

and

$$[\![I\ominus;\ zip_{F,\ominus}]\!]^{\cup} \circ \dagger.F.I \quad = \quad [\![I\ominus;\ zip_{F,\ominus}]\!]^{\cup}$$

but the latter follows from the domain property (93) of $zip_{F,\dagger}$. This leaves us:

$$\begin{array}{ll}
& [\![I\ominus;\ zip_{F,\ominus}]\!]^{\cup} \circ \dagger.\hat{F}.R \quad = \quad \hat{F}.\dagger.R \\
\equiv & \qquad \{ \text{ induction hypothesis: } zip_{F,\ominus} \quad = \quad (zip_{\ominus,F})^{\cup} \ \} \\
& [\![I\ominus;\ (zip_{\ominus,F})^{\cup}]\!]^{\cup} \circ \dagger.\hat{F}.R \quad = \quad \hat{F}.\dagger R \\
\equiv & \qquad \{ \text{ definition anamorphism } \} \\
& (\![I\ominus;\ zip_{\ominus,F}]\!) \circ \dagger.\hat{F}.R \quad = \quad \hat{F}.\dagger R \\
\equiv & \qquad \{ \text{ map fusion } \} \\
& (\![I\ominus;\ zip_{\ominus,F} \circ \hat{F}.R \ominus I]\!) \quad = \quad \hat{F}.\dagger R \\
\equiv & \qquad \{ \text{ uep } \} \\
& zip_{\ominus,F} \circ \hat{F}.R \ominus I \circ I \ominus \hat{F}.\dagger.R \quad = \quad \hat{F}.\dagger.R \quad \wedge \\
& \qquad \hat{F}.\dagger.R \circ \dagger.I \quad = \quad \hat{F}.\dagger.R
\end{array}$$

The second conjunct follows from (20) and (22), for the first conjunct we have:

$$\begin{array}{ll}
& zip_{\ominus,F} \circ \hat{F}.R \ominus I \circ I \ominus \hat{F}.\dagger.R \\
= & \qquad \{ \text{ generator fusion } \} \\
& zip_{\ominus,F} \circ \hat{F}.R \ominus \hat{F}.\dagger.R \\
= & \qquad \{ \text{ (31) } \} \\
& \hat{F}.(R \ominus \dagger.R) \\
= & \qquad \{ \text{ computation rule map } \} \\
& \hat{F}.\dagger.R \quad .
\end{array}$$

□

Solution to exercise: 113
Property (b) we verify by:

$$\begin{array}{ll}
& G.zip_{F,H} \circ zip_{F,G} \circ F.G.H.R \\
= & \qquad \{ \text{ (d) } \} \\
& G.zip_{F,H} \circ G.F.H.R \circ zip_{F,G} \\
= & \qquad \{ \text{ relators, (f) } \} \\
& G.H.F.R \circ G.zip_{F,H} \circ zip_{F,G} \quad .
\end{array}$$

For (c) we have:

$$\begin{array}{ll} & G.zip_{F,H} \circ zip_{F,G} \circ F.G.H.I \\ = & \quad \{ \text{ (d) } \} \\ & G.zip_{F,H} \circ G.F.H.I \circ zip_{F,G} \\ = & \quad \{ \text{ relators, (g) } \} \\ & G.zip_{F,H} \circ zip_{F,G} \quad . \end{array}$$

For (h) we have to prove

$$zip_{G\bullet H,+} \quad = \quad (zip_{+,G\bullet H})^{\cup}$$

but this follows from exercise 66 if we can prove:

$$(zip_{+,G\bullet H})^{\cup} \circ G.H.(R \blacktriangledown S) \quad = \quad G.H.R \blacktriangledown G.H.S$$

$$(zip_{+,G\bullet H})^{\cup} \circ G.H.(I{+}I) \quad = \quad (zip_{+,G\bullet H})^{\cup}$$

The same holds for the proof of (i). In general we have to prove:

$$(zip_{F,G\bullet H})^{\cup} \circ G.H.\hat{F}.R \quad = \quad \hat{F}.G.H.R$$

$$(zip_{F,G\bullet H})^{\cup} \circ G.H.F.I \quad = \quad (zip_{F,G\bullet H})^{\cup}$$

but the latter follows from (b) and (c). This leaves us:

$$\begin{array}{ll} & (zip_{F,G\bullet H})^{\cup} \circ G.H.\hat{F}.R \\ = & \quad \{ \text{ (a), induction hypothesis } \} \\ & zip_{G,F} \circ G.zip_{H,F} \circ G.H.\hat{F}.R \\ = & \quad \{ \text{ relators, (13) } \} \\ & zip_{G,F} \circ G.\hat{F}.H.R \\ = & \quad \{ \text{ (13) } \} \\ & \hat{F}.G.H.R \quad . \end{array}$$

The verification of (j) is more complicated. We start with

$$\begin{array}{ll} & zip_{\dagger,G\bullet H} \\ = & \quad \{ \text{ (a), induction hypothesis } \} \\ & G.(zip_{H,\dagger})^{\cup} \circ (zip_{G,\dagger})^{\cup} \\ = & \quad \{ \text{ (79), anamorphisms } \} \\ & G.(\!|I\ominus;\ (zip_{H,\ominus})^{\cup}|\!) \circ (\!|I\ominus;\ (zip_{G,\ominus})^{\cup}|\!) \\ = & \quad \{ \text{ induction hypothesis } \} \\ & G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \circ (\!|I\ominus;\ zip_{\ominus,G}|\!) \end{array}$$

Using this we have:

$$
\begin{array}{ll}
& zip_{\dagger,G\bullet H} \;=\; (zip_{G\bullet H,\dagger})^{\cup} \\
\equiv & \qquad \{ \text{ above, (79) } \} \\
& G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \circ (\!|I\ominus;\ zip_{\ominus,G}|\!) \;=\; (\!|I\ominus;\ (zip_{G\bullet H,\ominus})^{\cup}|\!) \\
\equiv & \qquad \{ \text{ induction hypothesis, (a) } \} \\
& G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \circ (\!|I\ominus;\ zip_{\ominus,G}|\!) \;=\; (\!|I\ominus;\ G.zip_{\ominus,H} \circ zip_{\ominus,G}|\!) \\
\Leftarrow & \qquad \{ \text{ fusion } \} \\
& G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \circ zip_{\ominus,G} \;=\; \\
& \qquad G.zip_{\ominus,H} \circ zip_{\ominus,G} \circ I\ominus G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \\
\Leftarrow & \qquad \{ \text{ computation rule, relator } G \} \\
& G.(I\ominus (\!|I\ominus;\ zip_{\ominus,H}|\!)) \circ zip_{\ominus,G} \;=\; zip_{\ominus,G} \circ I\ominus G.(\!|I\ominus;\ zip_{\ominus,H}|\!) \\
\Leftarrow & \qquad \{ G.(R\ominus S) \circ zip_{\ominus,G} \;=\; zip_{\ominus,G} \circ G.R\ominus G.S\ , \\
& \qquad\quad zip_{\ominus,G} \;=\; zip_{\ominus,G} \circ G.I\ominus G.I \ \} \\
& true\ .
\end{array}
$$

□

Solution to exercise: 114
Given is a natural isomorphism γ between F and G, i.e. we have:

(a) $F.R \circ \gamma \;=\; \gamma \circ G.R$ for all specs$R$,

(b) $F.I \;=\; \gamma{<} \;\wedge\; \gamma{>} \;=\; G.I$,

(c) γ is a bijection.

We try to calculate the generator for relator F by "forcing (20)":

$$
\begin{array}{ll}
& \hat{F}.(R \circ S) \;=\; F.R \circ \hat{F}.S \\
\equiv & \qquad \{ \bullet\ \hat{F}.S \;=\; \gamma \circ \hat{G}.S \ \} \\
& \gamma \circ \hat{G}.(R \circ S) \;=\; F.R \circ \gamma \circ \hat{G}.S \\
\equiv & \qquad \{ \text{ (a) } \} \\
& \gamma \circ \hat{G}.(R \circ S) \;=\; \gamma \circ G.R \circ \hat{G}.S \\
\equiv & \qquad \{ \text{ generator fusion } \} \\
& true\ .
\end{array}
$$

Thus we take $\hat{F}.R \;=\; \gamma \circ \hat{G}.R$.
The proof of (22) is trivial; for property (21) we have:

$$
\begin{array}{ll}
 & \hat{F}.\top\!\top \\
= & \quad \{ \ \text{definition } \hat{F} \ \} \\
 & \gamma \circ \hat{G}.\top\!\top \\
= & \quad \{ \ (21) \ \} \\
 & \gamma \circ G.\top\!\top \circ \top\!\top \\
= & \quad \{ \ (a) \ \} \\
 & F.\top\!\top \circ \gamma \circ \top\!\top \\
= & \quad \{ \ (b), \text{domains} \ \} \\
 & F.\top\!\top \circ \top\!\top \ .
\end{array}
$$

Next we want to construct $zip_{H,F}$ such that

$$zip_{H,F} \circ H.\hat{F}.R \quad = \quad \hat{F}.H.R \ .$$

We begin with the lefthand side and try to transform it to the right.

$$
\begin{array}{ll}
 & zip_{H,F} \circ H.\hat{F}.R \\
= & \quad \{ \ \text{definition } \hat{F}, \text{ relator } H \ \} \\
 & zip_{H,F} \circ H.\gamma \circ H.\hat{G}.R \\
= & \quad \{ \ \bullet \ zip_{H,F} \ = \ \alpha \circ H.\gamma^{\cup} \ \} \\
 & \alpha \circ H.(\gamma^{\cup} \circ \gamma) \circ H.\hat{G}.R \\
= & \quad \{ \ (b), (c)\text{: } \gamma^{\cup} \circ \gamma \ = \ G.I, \text{ generator fusion } \} \\
 & \alpha \circ H.\hat{G}.R \\
= & \quad \{ \ \bullet \ \alpha \ = \ \beta \circ zip_{H,G} \ \} \\
 & \beta \circ \hat{G}.H.R \\
= & \quad \{ \ \bullet \ \beta \ = \ \gamma \ \} \\
 & \hat{F}.H.R \ .
\end{array}
$$

Thus we take

$$zip_{H,F} \quad = \quad \gamma \circ zip_{H,G} \circ H.\gamma^{\cup} \ .$$

Now we verify the naturality property (14) for $zip_{H,F}$:

$$
\begin{array}{ll}
 & \gamma \circ zip_{H,G} \circ H.\gamma^{\cup} \circ H.F.R \\
= & \quad \{ \ \text{relator } H, \text{ (a) } \} \\
 & \gamma \circ zip_{H,G} \circ H.G.R \circ H.\gamma^{\cup} \\
= & \quad \{ \ \text{naturality property (14) for } zip_{H,G} \ \} \\
 & \gamma \circ G.H.R \circ zip_{H,G} \circ H.\gamma^{\cup} \\
= & \quad \{ \ (a) \ \} \\
 & F.H.R \circ \gamma \circ zip_{H,G} \circ H.\gamma^{\cup} \ .
\end{array}
$$

The domain property (15) is proved in a similar way:

$$
\begin{array}{ll}
 & \gamma \circ \mathit{zip}_{H,G} \circ H.\gamma^{\cup} \circ H.F.I \\
= & \qquad \{ \ \text{relator } H, \text{(a)} \ \} \\
 & \gamma \circ \mathit{zip}_{H,G} \circ H.G.I \circ H.\gamma^{\cup} \\
= & \qquad \{ \ \mathit{zip}_{H,G} \circ H.G.I \ = \ \mathit{zip}_{H,G} \ \} \\
 & \gamma \circ \mathit{zip}_{H,G} \circ H.\gamma^{\cup} \ .
\end{array}
$$

□

Solution to exercise: 123
Property (f) we prove by

$$
\begin{array}{ll}
 & \hat{F}.\check{G}.R \\
= & \qquad \{ \ \text{generator fusion} \ \} \\
 & F.\check{G}.I \circ \hat{F}.G.R \\
= & \qquad \{ \ \text{(e)} \ \} \\
 & F.\check{G}.I \circ \gamma \circ G.\hat{F}.R \\
= & \qquad \{ \ \text{(d)} \ \} \\
 & \check{G}.F.I \circ G.\hat{F}.R \\
= & \qquad \{ \ \text{generator fusion} \ \} \\
 & \check{G}.\hat{F}.R \ .
\end{array}
$$

For property (g) we have:

$$
\begin{array}{ll}
 & \hat{F}.(\!|G;\ R|\!) \ = \ (\!|G;\ F.R \circ \gamma|\!) \\
\equiv & \qquad \{ \ \text{uep} \ \} \\
 & \hat{F}.(\!|G;\ R|\!) \ = \ F.R \circ \gamma \circ G.\hat{F}.(\!|G;\ R|\!) \ \wedge \\
 & \qquad \hat{F}.(\!|G;\ R|\!) \circ \mu G \ = \ \hat{F}.(\!|G;\ R|\!)
\end{array}
$$

The second conjunct follow from (22). For the first we have:

$$
\begin{array}{ll}
 & F.R \circ \gamma \circ G.\hat{F}.(\!|G;\ R|\!) \\
= & \qquad \{ \ \text{(e)} \ \} \\
 & F.R \circ \hat{F}.G.(\!|G;\ R|\!) \\
= & \qquad \{ \ \text{generator fusion} \ \} \\
 & \hat{F}.(R \circ G.(\!|G;\ R|\!)) \\
= & \qquad \{ \ \text{computation rule} \ \} \\
 & \hat{F}.(\!|G;\ R|\!) \ .
\end{array}
$$

We calculate the dual property of (g):

$$
\begin{array}{ll}
& \forall(R :: \ \hat{F}.(\!|G;\ R|\!) \ = \ (\!|G;\ F.R \circ \gamma|\!)\,) \\
\equiv & \qquad \{ \ \text{reverse} \ \} \\
& \forall(R :: \ \check{F}.((\!|G;\ R|\!)^{\cup}) \ = \ (\!|G;\ F.R \circ \gamma|\!)^{\cup}\,) \\
\equiv & \qquad \{ \ \text{definition anamorphism} \ \} \\
& \forall(R :: \ \check{F}.[\![G;\ R^{\cup}]\!] \ = \ [\![G;\ \gamma^{\cup} \circ F.R^{\cup}]\!]\,) \\
\equiv & \qquad \{ \ \text{dummy change: } R \mapsto R^{\cup} \ \} \\
& \forall(R :: \ \hat{F}.[\![G;\ R]\!] \ = \ [\![G;\ \gamma^{\cup} \circ F.R]\!]\,) \ .
\end{array}
$$

□

Solution to exercise: 124
Taking $F := +$ (and adjusting for the fact that $+$ is a binary relator) we get from (g) of exercise 123:

$$(\!|F;\ R|\!) \blacktriangledown (\!|F;\ S|\!) \ = \ (\!|F;\ R{+}S \circ F.{\hookrightarrow}^{\cup} \blacktriangledown F.{\hookleftarrow}^{\cup}|\!) \ .$$

The dual property is:

$$[\![F;\ R]\!] \triangledown [\![F;\ S]\!] \ = \ [\![F;\ F.{\hookrightarrow} \triangledown F.{\hookleftarrow} \circ R{+}S]\!] \ .$$

□

Solution to exercise: 125
Taking $F := +$ and $G := +$ (and adjusting for the fact that both F and G are binary relators) it follows from exercise 123f that:

$$(R \triangledown S) \blacktriangledown (T \triangledown U) \ = \ (R \blacktriangledown T) \triangledown (S \blacktriangledown U)$$

Similarly, taking $F := \times$ and $G := \times$ we get:

$$(R \blacktriangle S) \vartriangle (T \blacktriangle U) \ = \ (R \vartriangle T) \blacktriangle (S \vartriangle U) \ .$$

□

Solution to exercise: 126
Property (c) is an instance of the theorem: if for spec R and S:

$$\hat{F}.R \circ S \ = \ \hat{F}.(R \circ S)$$

then

$$\hat{F}.G.R \circ G.S \quad = \quad \hat{F}.G.(R \circ S) \quad .$$

Proof

$$\begin{array}{ll} & \hat{F}.G.R \circ G.S \\ = & \qquad \{ \ (13) \ \} \\ & zip_{G,F} \circ G.\hat{F}.R \circ G.S \\ = & \qquad \{ \ \text{relator } G\text{, assumption} \ \} \\ & zip_{G,F} \circ G.\hat{F}.(R \circ S) \\ = & \qquad \{ \ (13) \ \} \\ & \hat{F}.G.(R \circ S) \quad . \end{array}$$

□

The generalised theorem for (d) is: if for spec R and S:

$$\hat{F}.R \circ S \quad = \quad \hat{F}.(R \circ S)$$

and

$\hat{G}.I$ is an imp

then

$$\hat{F}.G.R \circ \hat{G}.S \quad = \quad \hat{F}.\hat{G}.(R \circ S) \quad .$$

Proof

$$\begin{array}{ll} & \hat{F}.G.R \circ \hat{G}.S \\ = & \qquad \{ \ \text{generator fusion} \ \} \\ & \hat{F}.G.R \circ G.S \circ \hat{G}.I \\ = & \qquad \{ \ \text{theorem above} \ \} \\ & \hat{F}.G.(R \circ S) \circ \hat{G}.I \\ = & \qquad \{ \ \text{assumption: } \hat{G}.I \text{ is an imp, (22)} \ \} \\ & \hat{F}.(G.(R \circ S) \circ \hat{G}.I) \\ = & \qquad \{ \ \text{generator fusion} \ \} \\ & \hat{F}.\hat{G}.(R \circ S) \quad . \end{array}$$

□

References

[1] Maarten M. Fokkinga. *Law and Order in Algorithmics.* PhD thesis, Universiteit Twente, The Netherlands, 1992.