Polynomial relators

chnology

POLYNOMIAL RELATORS

by

Roland C. Backhouse
Paul Hoogendijk
Ed Voermans
Peter J. de Bruin
Grant Malcolm
Jaap van der Woude

Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands
All rights reserved
Editors: prof.dr.M.Rem
prof.dr.K.M. van Hee

POLYNOMIAL RELATORS
Draft

Roland C. Backhouse[*] Peter J. de Bruin[†] Paul Hoogendijk[*] Grant Malcolm[‡]
Ed Voermans[*] Jaap van der Woude[*][§]

May 28, 1991

Abstract

This paper reports ongoing research into a theory of datatypes based on the calculus of relations. A fundamental concept introduced here is the notion of "relator" which is an adaption of the categorical notion of functor. Axiomatisations of polynomial relators (that is relators built from the unit type and the disjoint sum and cartesian product relators) are given and their properties extensively catalogued. The effectiveness of the calculus is illustrated by the construction of several natural isomorphisms and natural simulations between relators.

[*]Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

[†]Department of Computer Science, Rijksuniversiteit Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands.

[‡]Computing Laboratory, Programming Research Group, Oxford University, 8-11 Keble Road, Oxford OX1 3QD, United Kingdom.

[§]CWI, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands.

1 Introduction

Research is underway into a theory of datatypes based on the calculus of relations. A fundamental concept within the theory is that of "relator" which takes the place of the notion of "functor" in a category-theory inspired development of functional programming.

In order that our theory be of any use we need to ensure that we can indeed define some non-trivial relators. It is well accepted that any useful theory of types should contain as a bare minimum the three components: a unit type, a disjoint sum operator and a cartesian product operator. In this paper we axiomatise these three within the calculus of relations and catalogue their properties. Following established terminology (see for example Manes and Arbib [11]) we call the relators so obtained the polynomial relators.

The main goal of our endeavour is to develop a calculus admitting compact and clear derivation of programs. To this end emphasis is placed on the identification of morphisms associated with each relator and their so-called "fusion" and "naturality" properties. Fusion properties characterise the circumstances under which a composition of relations, one of which is a morphism, can be combined into just one morphism (or vice-versa when a morphism can be "defused" into a composition of relations one of which remains a morphism). Naturality properties offer an alternative way of expressing monotonicity and fusion properties of morphisms; from a programming point of view their significance emerges when one tackles the problem of constructing simulations of one relator by another or isomorphisms between relators. The effectiveness of the calculus is demonstrated by exhibiting several such constructions.

A companion paper [3] provides more motivation for the development of the theory and takes it further into the domain of inductively-defined types.

We conclude this introduction with a short account of the style we use for presenting calculations.

Proof Format

For the presentation of equational proofs we use the style introduced by W.H.J. Feijen in [7]. That is, we write

$$\begin{array}{cl} & R \\ = & \quad \{ p \} \\ & S \\ = & \quad \{ q \} \\ & T \end{array}$$

R,S and T are expressions containing one or more free variables. p and q are most often semi-formal hints why (for all instantiations of the free variables) $R = S$ and $S = T$, respectively; in constructive proofs (discussed shortly) p and q have a formal status.

This format emphasises the transitivity of equality: all the expressions R, S and T are equal, but in particular the first and the last. We use other transitive operators in place of equality: $\equiv$ (equivalence), $\Leftarrow$ (follows from) $\Rightarrow$ (implies), $\sqsupseteq$ and $\sqsubseteq$ (the inclusion operators defined in section 3). In such cases the connectives are used *conjunctively*; for example $R \Leftarrow S \Leftarrow T$ means $(R \Leftarrow S)$ *and* $(S \Leftarrow T)$.

Peculiar to our own work is that we use the same proof style for *constructive* proofs. For example, we may wish to determine a condition q under which two given expressions R and T are equal. There are two ways we can proceed. One is to begin with the statement

$$R \;=\; T$$

and then in a series of steps derive q. Thus the derivation would take the form

$$\begin{array}{ll} & R \;=\; T \\ \Leftarrow & \quad \{ \text{ hint } \} \\ & \text{some intermediate steps} \\ \Leftarrow & \quad \{ \text{ hint } \} \\ & q \end{array}$$

Another way is to begin with R and try to transform it to T. On the way the conditions under which the transformation is possible are collected in the hints. Thus the proof takes the form

$$\begin{array}{ll} & R \\ = & \quad \{ q1 \} \\ & S \\ = & \quad \{ q2 \} \\ & T \end{array}$$

In such a proof the hints have a truly formal status and what is proven is the statement

$$q1 \wedge q2 \;\Rightarrow\; R = T$$

We forewarn the reader that this will be so by prefacing the proof with the words "by construction".

A particular case where such constructive proofs are used is the following. Given are two functions f and g and an expression R. Required is to find x such that $f.R = g.x$. I.e. we wish to prove the statement

$$\exists(x :: \; f.R \;=\; g.x)$$

This we often do by a stepwise refinement process in which, for reasons stated in the hints, we explore assignments to x of a particular form. The proof structure then takes a form something like:
By construction of x:

$$\begin{array}{ll} & f.R \;=\; g.x \\ \Leftarrow & \quad \{ \text{ reason why } f.R = g.(h.y) \Leftarrow f'.R = g'.y \;, \quad x := h.y \; \} \\ & f'.R \;=\; g'.y \\ \Leftarrow & \quad \{ \text{ reason why } f'.R = g'.T \; \} \\ & y := T \end{array}$$

Formally, such a proof establishes

$$\forall(x,y : \; x = h.y \;\wedge\; y = T \; : \; f.R \;=\; g.x)$$

which is of course equivalent to

$$f.R \;=\; g.(h.T)$$

The first example of such a proof can be found in section 9.

2 The Algebraic Framework

In this section we summarise relational algebra. For pedagogic reasons we decompose the algebra into three layers with interfaces between the layers plus two special axioms, one discussed here and one in a later section. The algebra is standard within the bounds of "scientific freedom".

2.1 Plat Calculus

Let $\mathcal{A}$ be a set, the elements of which are to be called *specs*. On $\mathcal{A}$ we impose the structure of a complete, completely distributive, complemented lattice

$$(\mathcal{A}, \sqcap, \sqcup, \neg, \top\!\!\top, \bot\!\!\bot)$$

where "$\sqcap$" and "$\sqcup$" are associative and idempotent, binary infix operators with unit elements "$\top\!\!\top$" and "$\bot\!\!\bot$", respectively, and "$\neg$" is the unary prefix operator denoting complement (or negation). We assume familiarity with the standard definition of a lattice given, for example, by Birkhoff [5]. By "complete lattice" we mean that the extremums

$$\sqcup(i : i \in \mathcal{I} : R_i)$$
and
$$\sqcap(i : i \in \mathcal{I} : R_i)$$

exist for all families of specs $\{i : i \in \mathcal{I} : R_i\}$, where the index set $\mathcal{I}$ is completely arbitrary. "Completely distributive lattice" means that

$$R \;\sqcap\; \sqcup(i : i \in \mathcal{I} : S_i) \;=\; \sqcup(i : i \in \mathcal{I} : R \sqcap S_i)$$
and
$$R \;\sqcup\; \sqcap(i : i \in \mathcal{I} : S_i) \;=\; \sqcap(i : i \in \mathcal{I} : R \sqcup S_i)$$

for all specs R and all families of specs $\{i : i \in \mathcal{I} : S_i\}$. Finally, "complemented lattice" means that $\neg R$ exists for all specs R and obeys de Morgan's laws and the double negation rule. (Note: the definition of a Boolean algebra requires only the existence of finite extremums and distributivity over such finite extremums. Our requirements are thus stronger.) The ordering relation induced by the lattice structure will be denoted by "$\sqsupseteq$".

This structure is well known from the predicate calculus: for "$\sqcap$" and "$\sqcup$" read conjunction and disjunction, respectively, for "$\top\!\!\top$" and "$\bot\!\!\bot$" read **true** and **false**, and for "$\sqsupseteq$" read "$\Leftarrow$". We call such a structure a *plat*, the "p" standing for power set and "lat" standing for lattice. Since the structure is so well known and well documented we shall assume a high degree of familiarity with it.

2.2 Composition

The second layer is the monoid structure for composition:

$$(\mathcal{A}, \circ, I)$$

where $\circ$ is an associative binary infix operator with unit element I.

The interface between these two layers is: $\circ$ is coordinatewise universally "cup-junctive". I.e. for $V, W \subseteq \mathcal{A}$,

$$(\sqcup V) \circ (\sqcup W) \;=\; \sqcup(P,Q : P \in V \wedge Q \in W : P \circ Q)$$

In particular,

- $\bot\!\!\bot$ is a left and right zero for $\circ$,
- $\circ$ is monotonic with respect to $\sqsupseteq$.
- $\top\!\!\top \circ \top\!\!\top \;=\; \top\!\!\top$.

2.3 Reverse

The third layer is the "reverse structure",

$$(\mathcal{A},\ {}^\cup)$$

where "${}^\cup$" is a unary postfix operator such that it is its own inverse.

The interface with the first layer is that "${}^\cup$" is an isomorphism of plats. I.e. for all $P, Q \in \mathcal{A}$,

$$P \sqsupseteq Q \quad \equiv \quad P^\cup \sqsupseteq Q^\cup$$

Consequently, for all $P, Q \in \mathcal{A}$,

$$\begin{array}{rcl} \neg(P^\cup) & = & (\neg P)^\cup \\ (P \sqcup Q)^\cup & = & P^\cup \sqcup Q^\cup \\ (P \sqcap Q)^\cup & = & P^\cup \sqcap Q^\cup \\ \top\!\!\top^\cup & = & \top\!\!\top \\ \bot\!\!\bot^\cup & = & \bot\!\!\bot \end{array}$$

Remark As a rule we shall write the names of unary functions as prefixes to their arguments. A partial justification for making an exception of "${}^\cup$" is that it commutes with "$\neg$", thus permitting us to write the syntactically ambiguous "$\neg R^\cup$". Later we shall see that "${}^\cup$" also commutes (by definition) with so-called "relators". The latter is the main reason for this choice of notation.
End of Remark

The interface with the second layer is that "${}^\cup$" is an isomorphism between the two monoid structures $(\mathcal{A},\ ;, I)$ and $(\mathcal{A},\ \circ,\ I)$ with

$$R\ ;\ S \quad = \quad S \circ R$$

In particular,

$$I^\cup \quad = \quad I$$

2.4 Operator precedence

Some remarks on operator precedence are necessary to enable the reader to parse our formulae. First, as always, operators in the metalanguage have lower precedence than operators in the object language. The principle meta-operators we use are equivalence ("$\equiv$"), implication ("$\Rightarrow$") and follows-from ("$\Leftarrow$") — these all having equal precedence — , together with conjunction ("$\wedge$") and disjunction ("$\vee$") — which have equal precedence higher than that of the other meta-operators. The precedence of the operators in the plat structure follows the same pattern. That is, "$=$", "$\sqsupseteq$" and "$\sqsubseteq$" all have equal precedence; so do "$\sqcup$" and "$\sqcap$"; and, the former is lower than the latter. Composition ("$\circ$") has a yet higher precedence than all of the operators mentioned thus far, whilst disjoint sum "$+$", cartesian product "$\times$" and their associated morphisms, junc "$\triangledown$" and split "$\vartriangle$" (all of which are introduced later) have the highest precedence of all the binary operators. Finally, all unary operators in the object language, whether prefix or postfix, have the same precedence which is the highest of all. Parentheses will be used to disambiguate expressions where this is necessary.

2.5 The RS and Rotation Rules

To the above axioms we now add an axiom that acts as an interface between all three layers.

The RS Rule

$$\neg Y \sqsupseteq P \circ \neg X \circ Q \quad \equiv \quad X \sqsupseteq P^{\cup} \circ Y \circ Q^{\cup}$$

The name "RS" is a mnemonic for "*R*otation and *S*hunting". The "rotation rule" is obtained by making the substitutions $Y := R^{\cup}$, $P := S$, $X := \neg T$ and $Q := I$ and simplifying using the properties of I, reverse and complement.

Rotation Rule

$$\neg R^{\cup} \sqsupseteq S \circ T \quad \equiv \quad \neg T^{\cup} \sqsupseteq R \circ S$$

(Note how the variables R, S and T are rotated in going from the left to the right side of the rule.) "Shunting" is the name given by Dijkstra and Scholten [8] to an important rule in the predicate calculus. Specifically, by making the substitutions $Y := U$, $P := I$, $X := V$, and $Q := W$ and simplifying we obtain the rule

$$\neg U \sqsupseteq \neg V \circ W \quad \equiv \quad V \sqsupseteq U \circ W^{\cup}$$

Interpreting "$\circ$" as conjunction, "$^{\cup}$" as the identity function, and "$\sqsupseteq$" as follows-from this is the afore-mentioned shunting rule.

It is our experience that the RS rule can meet with considerable resistance for one of two reasons. First, for calculational purposes, a rule with four free variables is (rightly) regarded as approaching, if not outwith, the limits of useability. Second, for those already familiar with the relational calculus, there is resistance to the fact that we have chosen to replace the better known "Schröder" rule which states that the following three statements are all equivalent.

$$\begin{array}{l} R \circ S \sqsubseteq T \\ R^{\cup} \circ \neg T \sqsubseteq \neg S \\ \neg T \circ S^{\cup} \sqsubseteq \neg R \end{array}$$

(See, for example, [16] for historical references.) To counter these arguments we would point out that the RS rule is more compact than the Schröder rule (two statements are equivalent rather than three) and, more importantly, has a clean syntactic form that makes it easy to remember and to apply. The rotation rule shares these advantages as well as involving only three free variables, but suffers the disadvantage that in some calculations two successive uses are required where only one use of the RS rule is necessary. In combination with other laws both rules are equivalent to the Schröder rule. (The Schröder rule can also be reduced to the equivalence of just two statements, making our first argument void, but then it would suffer the same disadvantage as the rotation rule, which is probably the reason why it is always stated in the way that it is.) An alternative axiomatisation is also possible using a yet simpler so-called "divergence rule" together with a rule relating factors to the complement operator. This alternative is discussed further in the appendix.

3 Foundations

The purpose of this section is to build up a vocabulary for our later discussion of the properties of morphisms. In order to avoid possible confusion with existing terminology we make a complete reappraisal of what is meant by "type", "function", "type constructor" etc. Nevertheless, it should be emphasised that — with the important exception of the notion of "relator" — *the concepts defined in this section, and their properties, are amply documented in the mathematical literature and we make no claim to originality.*

3.1 Monotypes

We say that spec A is a *monotype* iff $I \sqsupseteq A$.

In the relational model, for example, we may assume that the universe $\mathbb{U}$ contains two unequal values **true** and **false**. The monotype $\mathbb{B}$ of booleans is then defined to be the relation

$$\{(\mathbf{true}, \mathbf{true}), (\mathbf{false}, \mathbf{false})\}$$

Note that for monotypes A and B

(1) $\quad A \;=\; I \sqcap A \;=\; A^{\cup} \;=\; A \circ A$

(2) $\quad A \circ B \;=\; B \circ A \;=\; A \sqcap B$

Properties such as (1) and (2) stated here without proof are proven in the appendix.

We need to refer to the "domain" and "co-domain" (or "range") of a spec. In order to avoid unhelpful operational interpretations we use the terms *left-domain* and *right-domain* instead. These are denoted by "<" and ">", respectively, and defined by

(3) $\quad R< \;\hat{=}\; I \sqcap (R \circ R^{\cup})$

(4) $\quad R> \;\hat{=}\; I \sqcap (R^{\cup} \circ R)$

Since all the operators involved in their definition are monotonic it follows that "<" and ">" are monotonic. Relational calculus (see the appendix) yields the following alternative definitions defining $R<$ and $R>$ as the smallest monotypes satisfying the equations in A, $A \circ R = R$ and $R \circ A = R$, respectively.

(5) $\quad \forall(A : I \sqsupseteq A : A \circ R = R \;\equiv\; A \sqsupseteq R<)$

(6) $\quad \forall(A : I \sqsupseteq A : R \circ A = R \;\equiv\; A \sqsupseteq R>)$

Note that once again we choose to use a postfix notation for function application. On this occasion, however, it is *not* the case that complement and "<" (or ">") commute. That is $\neg(R<) \neq (\neg R)<$, in general. As we shall see, however, "<" and ">" do commute with relators and that is the reason for our choice.
The following properties of "<" also prove to be very useful. For all specs R and S,

(7) $\quad R< \;=\; (R^{\cup})>$

(8) $\quad R< \circ S \;=\; R \circ \top\!\top \sqcap S$

(9) $\quad (R \circ S)< \;=\; (R \circ S<)<$

(10) $\quad R< \;\sqsupseteq\; (R \circ S)<$

(11) $\quad (R \sqcup S)< \;=\; R< \sqcup S<$

(12) $\quad (R \sqcap S)< \;=\; R \circ S^{\cup} \sqcap I$

For convenience we also list the dual properties of ">".

(13) $\quad R> \;=\; (R^{\cup})<$

(14) $\quad S \circ R> \;=\; \top\!\top \circ R \sqcap S$

(15) $(R \circ S)> \;=\; (R> \circ S)>$

(16) $S> \;\sqsupseteq\; (R \circ S)>$

(17) $(R \sqcup S)> \;=\; R> \sqcup S>$

(18) $(R \sqcap S)> \;=\; R^{\cup} \circ S \sqcap I$

Of these six pairs of properties, five are evident when specs are interpreted as relations. One pair, properties (8) and (14), is less so. Nevertheless, it is worth drawing attention to them because they figure frequently in some of our calculations. The alternative expressions $I \sqcap R \circ \top\!\!\top$ and $I \sqcap \top\!\!\top \circ R$ for $R<$ and $R>$, respectively, are obtained from them by instantiating S to I and simplifying. In addition, the following distributivity properties are exploited surprisingly frequently.

(19) $(R \sqcap S \circ \top\!\!\top) \circ T \;=\; R \circ T \sqcap S \circ \top\!\!\top$

(20) $R \circ (S \sqcap \top\!\!\top \circ T) \;=\; R \circ S \sqcap \top\!\!\top \circ T$

Their proofs involve two applications of (8) or (14), as the case may be, together with associativity of composition.

We sometimes write

$$R \in S{\sim}T$$

as a synonym for

(21) $S \circ R \;=\; R \;=\; R \circ T$

It is immediate from (5) and (6) that

(22) $R< \circ R \;=\; R \;=\; R \circ R>$

Indeed this law is used so frequently that, after a while, we hardly bother to mention it. Using the notation we have just introduced (22) can be rephrased in the form

$$R \in R<{\sim}R>$$

Note that (21) defines $S{\sim}T$ to be a subset of $\mathcal{A}$. Typically S and T will be monotypes, but we prefer not to complicate the definition by making such a restriction.

It follows immediately from (2) with B instantiated to A that, for all monotypes A,

(23) $A \in A{\sim}A$

and, more specifically,

(24) $A< \;=\; A \;=\; A>$

Properties (7), (13) and (24) together with the properties of reverse (in particular, that it is its own inverse) have the important notational consequence that any sequence of applications of the left-/right- domain operators and/or the reverse operator can be reduced to the application of at most one of these operators. Such simplifications will be made automatically in our proofs except in one or two places where we judge that, in combination with the application of some other rule, the proof step has become too large for human consumption.

3.2 Imps and Co-imps

In this subsection we define "imps" and "co-imps" as special classes of specs. In the relational model an "imp" is a function.

Definition 25

(a) A spec f is said to be an *imp* if and only if $I \sqsupseteq f \circ f^{\cup}$.

(b) A spec f is said to be a *co-imp* if and only if $f^{\cup}$ is an imp.

(c) A spec is said to be a *bijection* if and only if it is both an imp and a co-imp.

□

We shall say that f is a bijection *between A and B* if it is a bijection and $f< = A$ and $f> = B$. Note that if this is the case then both A and B are monotypes and $A = f \circ f^{\cup}$ and $B = f^{\cup} \circ f$. The notation "$A \cong B$" (read as A is *isomorphic* to B) signifies the existence of a bijection between A and B.

Theorem 26 Composition preserves imps, co-imps and bijections.

Proof Straightforward.
□

The intended interpretation is that an "imp" is an "imp"lementation. On the other hand, it is not the intention that all implementations are "imps". Apart from their interpretation imps have an important distributive property not enjoyed by arbitrary specs, namely:

Theorem 27 If f is an imp then, for all non-empty sets of specs V,

$$\sqcap(P:\ P \in V:\ P) \circ f \quad = \quad \sqcap(P:\ P \in V:\ P \circ f)$$

In particular, for all specs R and S,

$$(R \sqcap S) \circ f \quad = \quad (R \circ f) \sqcap (S \circ f)$$

□

Dually we have:

Theorem 28 If f is a co-imp then, for all non-empty sets of specs V,

$$f \circ \sqcap(P:\ P \in V:\ P) \quad = \quad \sqcap(P:\ P \in V:\ f \circ P)$$

In particular, for all specs R and S,

$$f \circ (R \sqcap S) \quad = \quad (f \circ R) \sqcap (f \circ S)$$

□

Monotypes are examples of bijections. In the relational model a monotype is the identity function on that type. More generally, the requirement of being a function is the requirement of being single-valued on some subset of $\mathbb{U}$, the so-called "domain" of the function. The domain and range are made explicit in the following.

Definition 29 For monotypes A and B we define the set $A \longleftarrow B$ by $f \in A \longleftarrow B$ whenever

(a) $\quad A \sqsupseteq f \circ f^{\cup}$ $\quad$ and
(b) $\quad f{>} = B$

The nomenclature "$f \in A \longleftarrow B$" is verbalised by saying that "*f is an imp to A from B*".
□

In terms of the relational model, property (29a) expresses the statement that f is *zero-* or *single-valued*, i.e. for each x there is at most one y such that $y \langle f \rangle x$, and has range A. Property (29b) expresses the statement that f is *total* on domain B, i.e. for each $x \in B$ there is at least one y such that $y \langle f \rangle x$. Their combination justifies writing "$f.x$", for each $x \in B$, denoting the unique object y in A such that $y \langle f \rangle x$.

By including the above definition and not simultaneously including a dual notion for co-imps we have introduced an asymmetry into our theory that until now has been totally absent. This expresses a slight bias with an eye to the extension of the theory with cartesian product and disjoint sum. We hasten to add, nonetheless, that there is no such asymmetry in the theory at this instant and every property we state for imps alone has a dual property for co-imps.

It is easy to show that,

$$A \sim B \;\supseteq\; A \longleftarrow B$$

and, for imp f,

$$f \in f{<} \longleftarrow f{>}$$

as one would expect from the intended interpretations of these operators.

Note also that

$$f \in A \longleftarrow B \quad \Rightarrow \quad f \circ C \in A \longleftarrow (B \sqcap C)$$

In the case that $B \sqsupseteq C$, the imp $f \circ C$ is the restriction of f to domain C. A major advantage of viewing monotypes as specs is that type considerations can be readily incorporated into the calculations in this way. (For some examples see [10].)

We should stress that the two set-forming operations "$\sim$" and "$\longleftarrow$" do *not* form an essential part of our theory but are included in order that the reader may relate their existing knowledge of type structures to the present theory. In the sequel we shall often state properties of the domain-forming operations "<" and ">" and immediately transcribe them into properties of "$\sim$" and/or "$\longleftarrow$". We prefer the statements about the domains for two reasons: they offer a better separation of concerns and are thus calculationally more useful, and they can be stated with fewer dummies (and indeed in some cases with no dummies, although we don't go that far).

To avoid repeating assumptions and to assist the reader's understanding we continue to use the conventions that capital letters $A, B, C, \ldots$ at the beginning of the alphabet denote monotypes, small letters $f, g, h, \ldots$ denote imps or co-imps, and capital letters $R, S, T, \ldots$ at the end of the alphabet denote arbitrary specs.

Finally, let us remark that the unconventional direction of the arrow in the statement "$f \in A \longleftarrow B$" is entirely dictated by the choice to denote function application with the function name to the left of its argument. (We owe the suggestion to deviate from convention to Meertens [13].)

3.3 Relators

In categorical approaches to type theory a parallel is drawn between the notion of type constructor and the categorical notion of "functor", thereby emphasising that a type constructor is not just a function from types to types but also comes equipped with a function that maps arrows to arrows. For an informative account of this parallel see, for example, [11]. In this subsection we propose a modest extension to the notion of functor to which we give the name "relator".

Definition 30 A *relator* is a function, F, from specs to specs such that

(a) $I \sqsupseteq F.I$

(b) $R \sqsupseteq S \;\Rightarrow\; F.R \sqsupseteq F.S$

(c) $F.(R \circ S) \;=\; F.R \circ F.S$

(d) $F.(R^{\cup}) \;=\; (F.R)^{\cup}$

□

In view of (30d) we take the liberty of writing simply "$F.R^{\cup}$" without parentheses, thus avoiding explicit use of the property.

The above ostensibly defines a *unary* relator but we also wish to allow it to serve as the definition of a relator mapping an m-ary *vector* of specs into an n-ary *vector* of specs, for some natural numbers m and n. (This is necessary in order to allow the theory to encompass what are variously called "mutually recursive type definitions" and "many-sorted algebras". More generally, there is no reason why "m" and "n" may not be some fixed but nevertheless arbitrary index sets. However, such a generalisation would complicate the current discussion more than we deem justified.) The mechanism by which we can do this is to assume that all the constants appearing in the definition ("$=$", "$\sqsupseteq$", "I", "$\circ$" and "$\cup$") are silently "lifted" to operate on vectors. For example, if F maps m-ary vectors into n-ary vectors, property (30c) would be written out in the form

$$(F.(R_1 \circ S_1, \ldots, R_m \circ S_m))_j \;=\; (F.(R_1, \ldots, R_m))_j \circ (F.(S_1, \ldots, S_m))_j$$

for all j, $1 \le j \le n$, whereby the use of subscripts denotes projection of a vector onto one of its components. It is, however, just such clumsy expressions that we want to avoid.

There are two cases that we make particular use of. The first case has to do with taking fixed points of relators where an obvious requirement is that the arity of the domain vector of the relator is identical to that of its range vector. We call such a relator an *endorelator* in conformance with the teminology "endofunctor" used in category theory. The second case is when F maps a pair of (vectors of) specs into a (vector of) spec(s). We refer to such relators as *binary* relators and choose to denote them by infix operators. Thus, if $\otimes$ denotes a binary relator, its defining properties would be spelt out as follows.

(a) $I \sqsupseteq I \otimes I$

(b) $R \sqsupseteq S \wedge U \sqsupseteq V \;\Rightarrow\; R \otimes U \sqsupseteq S \otimes V$

(c) $(R \circ S) \otimes (U \circ V) \;=\; (R \otimes U) \circ (S \otimes V)$

(d) $(R^{\cup}) \otimes (S^{\cup}) \;=\; (R \otimes S)^{\cup}$

The notational advantage of writing "$\cup$" as a postfix to its argument is, of course, lost in this case.

A property such as (c) we call an "abide" law; we also often refer to this law by saying that binary relators "abide" with composition. The name was coined by Richard Bird (in a different context). His motivation for the name was that it is short for "above/beside" reflecting the following two-dimensional formulation of the law in which the relator and composition are either above or beside each other.

$$\begin{array}{ccccccc} R & \circ & S & & R & & S \\ & \otimes & & = & \otimes & \circ & \otimes \\ U & \circ & V & & U & & V \end{array}$$

(Our first encounter with a two-dimensional depiction of an abide law was in [9]. In the category theory literature the term "interchange" rule (or law) is used.)

As already announced relators commute with the domain operators.

Theorem 31 If F is a relator then

(a) $\quad F.(R>) \;=\; (F.R)>$

(b) $\quad F.(R<) \;=\; (F.R)<$

□

For the proof of this theorem see [2].

In view of theorem 31 we write "$F.R<$" and "$F.R>$" without parentheses, again in order to avoid explicit mention of the properties.

The following theorem allows a comparison to be made with our definition of "relator" and the definition of "functor" (in the category of sets).

Theorem 32 If F is a relator then

(a) $\quad A$ is a monotype $\Rightarrow$ $F.A$ is a monotype

(b) $\quad f$ is an imp $\Rightarrow$ $F.f$ is an imp

(c) $\quad f$ is a co-imp $\Rightarrow$ $F.f$ is a co-imp

(d) $\quad f \in A \longleftarrow B \;\Rightarrow\; F.f \in F.A \longleftarrow F.B$

(e) $\quad R \in A \sim B \;\Rightarrow\; F.R \in F.A \sim F.B$

Proof
Straightforward instantiation of the definitions of "monotype", "imp", "co-imp", "$\longleftarrow$" and "$\sim$" combined with the definition of a relator and, in the case of part (b), theorem 31.
□

3.4 Cap- and Cup-Junctivity

In addition to the four defining properties of a relator one might ask the question whether it distributes over the cup and/or the cap operator. Such a property we call a "finite junctivity" property. More generally, one might ask whether the relator distributes over some class of quantifications with respect to the cup and/or cap operator. In order to make the latter notion precise we introduce the following definition.

Definition 33 Suppose $\mathcal{I}$ is a set. We use i and j to denote elements of $\mathcal{I}$. An $\mathcal{I}$-*bag* is a (total) spec-valued function with domain $\mathcal{I}$. If $\mathcal{R}$ is an $\mathcal{I}$-bag then $\mathcal{R}.i$ denotes the spec obtained by applying $\mathcal{R}$ to $i \in \mathcal{I}$. Also $\sqcup_{\mathcal{I}}\mathcal{R}$ is used to denote $\sqcup(i : i \in \mathcal{I} : \mathcal{R}.i)$.

An $\mathcal{I}$-bag, $\mathcal{R}$, is *linear* if for all $i,j \in \mathcal{I}$ one has either $\mathcal{R}.i \sqsupseteq \mathcal{R}.j$ or $\mathcal{R}.j \sqsupseteq \mathcal{R}.i$.

We call a function G from specs to specs $\mathcal{I}$-*cup-junctive* if for all $\mathcal{I}$-bags, $\mathcal{R}$,

(34) $\quad G.(\sqcup_{\mathcal{I}}\mathcal{R}) \;=\; \sqcup_{\mathcal{I}}(G \bullet \mathcal{R})$

(Note: " $\bullet$ " denotes composition of functions.)

$\mathcal{I}$-cap-junctivity is defined similarly.

The function G is said to be *$\mathcal{I}$-cup-continuous* if (34) holds for all linear bags $\mathcal{R}$. The notion of *$\mathcal{I}$-cap-continuity* is similarly defined.

(For clarity it is as well to remark that the definition of $\mathcal{I}$-cap-junctivity for a *binary* operator $\otimes$ is that for each *pair* of $\mathcal{I}$-bags $\mathcal{R}$ and $\mathcal{S}$,

$$\sqcap(i : i \in \mathcal{I} : \mathcal{R}.i) \otimes \sqcap(i : i \in \mathcal{I} : \mathcal{S}.i) \quad = \quad \sqcap(i : i \in \mathcal{I} : \mathcal{R}.i \otimes \mathcal{S}.i)$$

Written without dummies this is the statement

$$\sqcap_{\mathcal{I}}\mathcal{R} \otimes \sqcap_{\mathcal{I}}\mathcal{S} = \sqcap_{\mathcal{I}}(\otimes \bullet \langle \mathcal{R}, \mathcal{S} \rangle)$$

A similar statement holds for $\mathcal{I}$-cup-junctivity.)

□

Using the word "junctive" to stand for both "cap-junctive" and "cup-junctive", and "continuous" to stand for "cap-continuous" and "cup-continuous" we may identify the following properties:

- universally junctive, i.e. junctive over all $\mathcal{I}$.
- positively junctive, i.e. junctive over all non-empty $\mathcal{I}$.
- denumerably junctive, i.e. junctive over all non-empty $\mathcal{I}$ with denumerably many elements.
- finitely junctive, i.e. junctive over all non-empty finite $\mathcal{I}$.
- continuous, i.e. junctive over all non-empty, linear $\mathcal{I}$.
- monotonic, i.e. junctive over all non-empty, finite, linear, $\mathcal{I}$.

The relationship between these various types of junctivity properties is discussed in some depth by Dijkstra and Scholten [8] (in the context of a plat calculus) from where our definitions are borrowed.

As examples of the use of this terminology, we would say that the functions $(R\circ)$ and $(\circ R)$ are (by postulate) universally cup-junctive for all specs R. Moreover (see theorems 27 and 28), $(f\circ)$ is positively cap-junctive for all co-imps f, and $(\circ f)$ is positively cap-junctive for all imps f.

We trust that it is evident why we are interested in properties such as continuity.

4 Natural Polymorphism

Any discussion of a theory of datatypes would be incomplete without a discussion of polymorphism. This is particularly true here because our theory is principally a theory of two sets of polymorphic functions — the relators and their associated morphisms. Relators are polymorphic in the sense that they may be applied to arbitrary specs irrespective of the domains of the argument spec. Such a statement is, however, somewhat banal since it says nothing about the mathematical nature of the claimed polymorphism. In this section we shall argue that relators are "naturally polymorphic". The latter notion is an adaptation and extension of the notion of "natural transformation" in category theory; the definition that we use is based on the work of de Bruin [6] which work was anticipated by Reynolds [14].

4.1 Higher-Order Spec Algebras

Expressing the natural polymorphism of relators (and other functions or relations) requires the notion of higher-order spec algebra which we now define.

Let SPEC $= (\mathcal{A},\ \sqsupseteq,\ I,\ \circ,\ ^{\cup})$ be a spec-algebra. Then the algebra of binary relations on specs $\overline{\text{SPEC}}$ is defined to be $(\overline{\mathcal{A}},\ \overline{\sqsupseteq},\ \overline{I},\ \overline{\circ},\ \overline{^{\cup}})$ where

$$\begin{array}{rcl} \overline{\mathcal{A}} & = & \mathbb{P}(\mathcal{A}\times\mathcal{A}) \\ \overline{\sqsupseteq} & = & \supseteq \end{array}$$

and, using the notation $x\,\langle R\rangle\,y$ instead of $(x,y)\in R$,

$$\begin{array}{rcl} x\,\langle \overline{I}\rangle\,y & \equiv & x = y \\ x\,\langle R\ \overline{\circ}\ S\rangle\,z & \equiv & \exists(y :: x\,\langle R\rangle\,y \wedge y\,\langle S\rangle\,z) \\ x\,\langle R^{\overline{\cup}}\rangle\,y & \equiv & y\,\langle R\rangle\,x \end{array}$$

for all $R,S \in \overline{\mathcal{A}}$ and all x,y and $z\in\mathcal{A}$. With these definitions, $\overline{\text{SPEC}}$ is also a spec algebra. We call $\overline{\text{SPEC}}$ a *higher-order spec algebra.*

The imps of $\overline{\text{SPEC}}$ are (partial) functions to $\mathcal{A}$ from $\mathcal{A}$. Specifically, the function f from $\mathcal{A}$ to $\mathcal{A}$ is identifed with the relation f on $\mathcal{A}\times\mathcal{A}$ where

$$x\,\langle f\rangle\,y \quad\equiv\quad x = f.y$$

for all $x,y\in\mathcal{A}$. Examples of imps in $\overline{\text{SPEC}}$ are the relators of SPEC. Note that relators are *total* imps. I.e. for each relator F we have

$$F^{\overline{\cup}}\ \overline{\circ}\ F \ \ \overline{\sqsupseteq}\ \ \overline{I}$$

The monotypes of $\overline{\text{SPEC}}$ can be identified with the subsets of $\mathcal{A}$. That is, a binary relation $\overline{A}$ in $\overline{\mathcal{A}}$ is a monotype if and only if there is an element A of $\mathbb{P}(\mathcal{A})$ such that

(35) $\quad \forall(x,y :: x\,\langle\overline{A}\rangle\,y \ \equiv\ x = y \wedge x\in A)$

The operators "$\sim$" and "$\longleftarrow$" were defined in section 3.2 as set-forming operators. Using (35) to identify monotypes of $\overline{\text{SPEC}}$ with subsets of $\mathcal{A}$, we may identify "$\sim$" and "$\longleftarrow$" with elements of $\overline{\mathcal{A}}$, specifically with binary relations on elements of $\mathcal{A}$ that are subsets of the identity relation $\overline{I}$. To reinforce this identification we corrupt the normal usage of the belongs-to symbol "$\in$" by the following definition. For spec R and relation $\overline{S}$ we define

$$R\ \overline{\in}\ \overline{S} \quad\equiv\quad R\langle\overline{S}\rangle R$$

Of course, $\overline{\text{SPEC}}$ can itself serve as the basis for the construction of a second algebra of binary relations $\overline{\overline{\text{SPEC}}}$, and in this way one can construct an infinite hierarchy of spec algebras. The relators and morphism constructors of one algebra are then total imps in the next higher order algebra; similarly, the expressions "$A \sim B$" and "$A \longleftarrow B$" of one algebra may be identified with monotypes in the next higher order algebra. Maintaining the distinction between the levels has been one reason why we have continually distinguished between "specs" and "relations", and between "imps" and "functions".

In this section we define three more relations which we call the naturality operators. The operators will be used at various levels in the hierarchy of SPEC algebras but we do not bother to decorate the different uses with a bar to indicate the level of use. Similarly, we use the undecorated symbols "$\longleftarrow$" , "$\sim$", "$\circ$", "$^{\cup}$" etc. at all levels in the hierarchy. The definitions of the barred operators given earlier will be important to reducing statements at one level to statements at the next lower level. Their use is, of course, only permitted within higher-order algebras.

As an example of this overloading of notation and in order to provide a reference point for our later discussion let us note the following properties:

Theorem 36 Let F be a relator. Then, for all monotypes A and B,

(a) $F \circ (A \sim B) \in (F.A \sim F.B) \longleftarrow (A \sim B)$

(b) $F \circ (A \longleftarrow B) \in (F.A \longleftarrow F.B) \longleftarrow (A \longleftarrow B)$

□

To understand these statements one must understand at what level each of the operators is being used. Theorem 36(a) is exemplary. Reintroducing the bar notation it states that

$$F \mathbin{\bar{\circ}} (A \sim B) \;\; \bar{\in} \;\; (F.A \sim F.B) \overline{\longleftarrow} (A \sim B)$$

Thus all operators are higher-order but for the "$\sim$" operators. Note that $F \mathbin{\bar{\circ}} (A \sim B)$ is the restriction of relator F to elements of $A \sim B$. A more conventional (but calculationally less convenient) notation might be $F_{A\sim B}$ (or $F_{A,B}$) indicating that relators are families of functions indexed by pairs of monotypes. Statement (b) is interpreted similarly; all operators are higher-order but for the first, second and fourth occurrences of "$\longleftarrow$".

Armed with this insight we may verify part (a) as follows.

$$\begin{array}{ll}
 & F \circ (A \sim B) \in (F.A \sim F.B) \longleftarrow (A \sim B) \\
\equiv & \qquad \{ \text{ definition of } \longleftarrow \ \} \\
 & P1 \wedge P2 \wedge P3
\end{array}$$

where

$$\begin{array}{lcl}
P1 & \equiv & F \circ (A \sim B) \;\; \text{is an imp} \\
P2 & \equiv & (F \circ (A \sim B))> \;=\; A \sim B \\
P3 & \equiv & (F.A \sim F.B) \;\sqsupseteq\; (F \circ (A \sim B))<
\end{array}$$

Property (P1) is immediate from the fact that relators are (higher-order) imps, as are monotypes, and composition of imps gives an imp. Here F is a relator and $A \sim B$ is —strictly, is identified with— a higher-order monotype. Also, P2 follows from the following calculation:

$$\begin{array}{ll}
 & (F \circ (A \sim B))^{\cup} \circ (F \circ (A \sim B)) \\
= & \qquad \{ \text{ properties of reverse } \} \\
 & (A \sim B)^{\cup} \circ F^{\cup} \circ F \circ (A \sim B) \\
\sqsupseteq & \qquad \{ \text{ relators are by definition total, i.e. } F^{\cup} \circ F \sqsupseteq I \ \} \\
 & (A \sim B)^{\cup} \circ (A \sim B) \\
= & \qquad \{ \ A \sim B \text{ is a monotype , (1) } \} \\
 & A \sim B
\end{array}$$

Finally,

$P3$

$\equiv$ { definition of left domain, properties of reverse }

$F.A \sim F.B \;\sqsupseteq\; I \sqcap F \circ (A \sim B) \circ F^{\cup}$

$\equiv$ { definition of higher-order operators }

$\forall(R :: R\langle F.A \sim F.B\rangle R \;\Leftarrow\; R\langle F \circ (A \sim B) \circ F^{\cup}\rangle R)$

$\equiv$ { definition of higher-order operators, predicate calculus }

$\forall(R : \exists(S :: (R = F.S) \wedge S\langle A \sim B\rangle S) : R\langle F.A \sim F.B\rangle R)$

$\equiv$ { predicate calculus, definition of $A \sim B$ }

$\forall(S : A \circ S = S = S \circ B : F.A \circ F.S = F.S = F.S \circ F.B)$

$\equiv$ { relators distribute through composition }

true

Although on the face of it quite long, the above proof simply involves unfolding the definitions of the higher-order operators and, in the last step, exploiting the distributivity of relators over composition. The property is, in part, just another way of stating theorem 32(c) but with one fewer bound variable. Property 36(b) may be verified in a similar way; it restates theorem 32(b).

4.2 The Naturality Operators

Saving one bound variable is hardly justification for such a spate of definitions. The motivation for presenting theorem 36 was to be able to compare it to theorem 39 below. First, yet three more definitions.

Definition 37 (The Naturality Operators) Let R and S be specs. Then we define the relations $R \leftsquigarrow S$, $R \rightsquigarrow S$ and $R \leftrightsquigarrow S$ by

(a) $U\langle R \leftsquigarrow S\rangle V \;\equiv\; R \circ V \sqsupseteq U \circ S$

(b) $U\langle R \rightsquigarrow S\rangle V \;\equiv\; R \circ V \sqsubseteq U \circ S$

and

(c) $U\langle R \leftrightsquigarrow S\rangle V \;\equiv\; R \circ V = U \circ S$

□

The above definition of the $\leftsquigarrow$ operator was introduced in [1]; it is related by part (a) of the following theorem to definitions introduced variously by deBruin [6], Reynolds [14] and Wadler [17].

Theorem 38

(a) If R and S are relations and f and g are total functions then
$$f\langle R \leftsquigarrow S\rangle g \;\equiv\; \forall(u,v :: f.u\langle R\rangle g.v \Leftarrow u\langle S\rangle v)$$
(b) If R and S are relations and $f^{\cup}$ and $g^{\cup}$ are total functions then
$$f\langle R \rightsquigarrow S\rangle g \;\equiv\; \forall(u,v :: u\langle R\rangle v \Rightarrow f^{\cup}.u\langle S\rangle g^{\cup}.v)$$
(c) If R and S are relations and f and g are total, surjective bijections then
$$f\langle R \leftrightsquigarrow S\rangle g \;\equiv\; \forall(u,v :: f.u\langle R\rangle g.v \equiv u\langle S\rangle v)$$
□

See [1] for the (straightforward) proof of part (a) of this theorem.

Several other more evident properties of these operators will be assumed in the sequel, an example being that $\leftsquigarrow$ is anti-monotonic in its second argument.

4.3 Naturality of Relators, Reverse and Composition

The reader is invited to compare the following theorem with theorem 36.

Theorem 39 (Naturality of Relators) If F is a relator then for all specs R and S

(a) $F \in (F.R \leftsquigarrow F.S) \leftsquigarrow (R \leftsquigarrow S)$

(b) $F \in (F.R \rightsquigarrow F.S) \leftsquigarrow (R \rightsquigarrow S)$

(c) $F \in (F.R \leftrightsquigarrow F.S) \leftsquigarrow (R \leftrightsquigarrow S)$

Proof The proof of part (a) proceeds as follows.

$$\begin{array}{ll}
 & F \in (F.R \leftsquigarrow F.S) \leftsquigarrow (R \leftsquigarrow S) \\
\equiv & \quad \{ \text{ definition of } \in \ \} \\
 & F\langle (F.R \leftsquigarrow F.S) \leftsquigarrow (R \leftsquigarrow S)\rangle F \\
\equiv & \quad \{ \text{ theorem 38, relators are total functions } \} \\
 & \forall (U,V :: F.U\langle F.R \leftsquigarrow F.S\rangle F.V \ \Leftarrow \ U\langle R \leftsquigarrow S\rangle V) \\
\equiv & \quad \{ \text{ definition 37(a) } \} \\
 & \forall (U,V :: F.R \circ F.V \sqsupseteq F.U \circ F.S \ \Leftarrow \ R \circ V \sqsupseteq U \circ S) \\
\equiv & \quad \{ \text{ relators distribute through composition and are monotonic } \} \\
 & \textbf{true}
\end{array}$$

The proofs of parts (b) and (c) are identical but for the replacement of the inclusion symbol by, respectively, the containment symbol and the equality symbol in the penultimate step.

□

Nowhere in this document do we hazard a definition of "natural polymorphism". Theorem 39 does, however, express precisely what we intend by the informal statement that relators are "naturally polymorphic". Similar theorems are proved later about the basic constituents of cartesian products and disjoint sums. In each case the theorem involves a universal quantification over specs, and it is in this sense that the spec in question is "polymorphic". The adjective "naturally" is added to suggest the link with "natural transformation" in category theory and to avoid confusion of our notion of polymorphism with existing notions.

There is, of course, much more to be said about the naturality operators. Statements such as theorem 39 express something about the "type" of specs, but along with a notion of type one would normally expect a notion of type inference. A first step to formulating such a type inference algorithm is the observation that composition is also naturally polymorphic. Specifically we have.

Theorem 40 (Naturality of Composition)
For all $\sim \ \in \{\leftsquigarrow, \rightsquigarrow, \leftrightsquigarrow\}$ and all specs P_1, P_2, Q_1, Q_2, R, S, T,

$$P_1 \circ Q_1 \langle R \sim T\rangle P_2 \circ Q_2 \ \Leftarrow \ P_1\langle R \sim S\rangle P_2 \ \wedge \ Q_1\langle S \sim T\rangle Q_2$$

In particular,

$$P \circ Q \in R \sim T \ \Leftarrow \ P \in R \sim S \ \wedge \ Q \in S \sim T$$

Proof Suppose $\sim \ \in \{\leftsquigarrow, \rightsquigarrow, \leftrightsquigarrow\}$. Let $\trianglelefteq$ denote $\sqsupseteq$, $\sqsubseteq$ or $=$ depending on the value of $\sim$. Then we have:

$$
\begin{array}{ll}
 & P_1 \langle R \mathbin{\dot{\sim}} S \rangle P_2 \quad \wedge \quad Q_1 \langle S \mathbin{\dot{\sim}} T \rangle Q_2 \\
\equiv & \qquad \{ \text{ definition of } \dot{\sim} \text{ } \} \\
 & R \circ P_2 \trianglelefteq P_1 \circ S \quad \wedge \quad S \circ Q_2 \trianglelefteq Q_1 \circ T \\
\Rightarrow & \qquad \{ \text{ composition is monotonic with respect to } \trianglelefteq \text{ } \} \\
 & R \circ P_2 \circ Q_2 \trianglelefteq P_1 \circ S \circ Q_2 \quad \wedge \quad P_1 \circ S \circ Q_2 \trianglelefteq P_1 \circ Q_1 \circ T \\
\Rightarrow & \qquad \{ \text{ transitivity of } \trianglelefteq \text{ } \} \\
 & R \circ P_2 \circ Q_2 \trianglelefteq P_1 \circ Q_1 \circ T \\
\equiv & \qquad \{ \text{ definition of } \dot{\sim} \text{ } \} \\
 & P_1 \circ Q_1 \langle R \mathbin{\dot{\sim}} T \rangle P_2 \circ Q_2
\end{array}
$$

The corollary is obtained by instantiating P_1 and P_2 to P and Q_1 and Q_2 to Q.
□

Remark We shall often use $\dot{\sim}$ as a (universally quantified) variable ranging over $\{\dot{\leftarrow}, \dot{\rightarrow}, \dot{\leftrightarrow}\}$ and $\trianglelefteq$ as a variable ranging over $\{\sqsupseteq, =, \sqsubseteq\}$. Sometimes we use them both simultaneously, as above, in which case they correspond (i.e. if $\dot{\sim}$ is $\dot{\leftarrow}$ then $\trianglelefteq$ is $\sqsupseteq$, if $\dot{\sim}$ is $\dot{\leftrightarrow}$ then $\trianglelefteq$ is $=$, and if $\dot{\sim}$ is $\dot{\rightarrow}$ then $\trianglelefteq$ is $\sqsubseteq$.) At other times they are used singly. In addition we sometimes use $\trianglerighteq$ in the rôle of a variable (always in combination with $\trianglelefteq$) in which case it designates the reverse of the relation designated by $\trianglelefteq$. *End of Remark*

To this we add the "naturally polymorphic type" of reverse. Unfortunately, our notation does not permit this to be done in a single statement. Instead, three are needed. Note the interchange of left- and right-pointing arrows in the first two.

Theorem 41 (Naturality of Reverse)

(a) $\quad \cup \in (R^{\cup} \mathbin{\dot{\leftarrow}} S^{\cup}) \mathbin{\dot{\leftrightarrow}} (S \mathbin{\dot{\rightarrow}} R)^{\cup}$
(b) $\quad \cup \in (R^{\cup} \mathbin{\dot{\rightarrow}} S^{\cup}) \mathbin{\dot{\leftrightarrow}} (S \mathbin{\dot{\leftarrow}} R)^{\cup}$
(c) $\quad \cup \in (R^{\cup} \mathbin{\dot{\leftrightarrow}} S^{\cup}) \mathbin{\dot{\leftrightarrow}} (S \mathbin{\dot{\leftrightarrow}} R)^{\cup}$

Proof We prove (a) as an example.

$$
\begin{array}{ll}
 & \cup \in (R^{\cup} \mathbin{\dot{\leftarrow}} S^{\cup}) \mathbin{\dot{\leftrightarrow}} (S \mathbin{\dot{\rightarrow}} R)^{\cup} \\
\equiv & \qquad \{ \text{ theorem 38(c), reverse is an isomorphism } \} \\
 & \forall(U,V :: \; U^{\cup} \langle R^{\cup} \mathbin{\dot{\leftarrow}} S^{\cup} \rangle V^{\cup} \;\equiv\; U \langle (S \mathbin{\dot{\rightarrow}} R)^{\cup} \rangle V) \\
\equiv & \qquad \{ \text{ definition 37(a), reverse and definition 37(b) } \} \\
 & \forall(U,V :: \; R^{\cup} \circ V^{\cup} \;\sqsupseteq\; U^{\cup} \circ S^{\cup} \;\equiv\; S \circ U \sqsubseteq V \circ R) \\
\equiv & \qquad \{ \text{ reverse } \} \\
 & \mathbf{true}
\end{array}
$$

□

5 The Unit Type

The unit type corresponds to a set with only one element; not a particularly interesting type, but nevertheless useful as a building block for constructing more complex data structures. The theory presented so far doesn't provide a vocabulary for talking about elements, only for talking about specs: this is not unintentional since a goal of our work has always been to minimise the incidence of point-wise arguments. In keeping with this goal, we adopt a rather abstract view of data types, and take a roundabout route to characterise the unit type.

5.1 The Cone Rule

We begin by postulating an axiom dubbed "the cone rule". This axiom could equally well have been included in section 2. It has been included here because it is only within the axiomatisation of the unit type that we make any use of the rule. Elsewhere (e.g. [16]) the cone rule is called "Tarski's rule."

The Cone Rule

$$\top\!\!\top \circ R \circ \top\!\!\top = \top\!\!\top \quad \equiv \quad R \neq \bot\!\!\bot$$

As a partial motivation for the cone rule we ask the reader to compare it with the following consequence of the RS rule.

Lemma 42 For all specs R, the following statements are all equivalent:

$$\begin{array}{lrcl}
\text{(a)} & \bot\!\!\bot & = & R \\
\text{(b)} & \bot\!\!\bot & = & R \circ \top\!\!\top \\
\text{(c)} & \bot\!\!\bot & = & \top\!\!\top \circ R \\
\text{(d)} & \bot\!\!\bot & = & \top\!\!\top \circ R \circ \top\!\!\top \\
\text{(e)} & \bot\!\!\bot & = & R< \\
\text{(f)} & \bot\!\!\bot & = & R>
\end{array}$$

Proof Suppose R is an arbitrary spec. Then, it is obvious that (a) implies both (b) and (c) (since $\bot\!\!\bot$ is a zero of composition). By the same token, each of (b) and (c) imply (d). To show that all of (a) to (d) are equivalent it suffices, therefore, to show that (d) implies (a). We actually show their equivalence, as follows:

$$\begin{array}{ll}
 & \bot\!\!\bot \;=\; \top\!\!\top \circ R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ plat calculus } \} \\
 & \bot\!\!\bot \;\sqsupseteq\; \top\!\!\top \circ R \circ \top\!\!\top \\
\equiv & \quad \{ \text{ RS rule } \} \\
 & \neg R \;\sqsupseteq\; \top\!\!\top^{\cup} \circ \neg\bot\!\!\bot \circ \top\!\!\top^{\cup} \\
\equiv & \quad \{ \text{ elementary properties of SPEC algebra } \} \\
 & \neg R \;\sqsupseteq\; \top\!\!\top \\
\equiv & \quad \{ \text{ plat calculus } \} \\
 & R \;=\; \bot\!\!\bot
\end{array}$$

Finally, (e) is equivalent to (b) and (f) to (c) because the righthand sides of each pair of equations are equal (by (8) and (14)).
□

One consequence of the cone rule is that $\top\!\!\top$ and $\bot\!\!\bot$ are different. More significantly, by combining the cone rule and lemma 42, one sees that the spec $\top\!\!\top \circ R \circ \top\!\!\top$ is always either $\top\!\!\top$ or $\bot\!\!\bot$ whatever the value of spec R. We say that the function mapping R to $\top\!\!\top \circ R \circ \top\!\!\top$ is *boolean-valued*; the cone rule itself is an abstract and concise way of expressing the proposition that, considered as a set of pairs, spec R either contains no elements or contains at least one element.

Another consequence of the cone rule, that we mention for later use, is the following:

Lemma 43 $\qquad \bot\!\!\bot = R \circ \top\!\!\top \circ S \quad \equiv \quad \bot\!\!\bot = R \vee \bot\!\!\bot = S$

Proof

$$\begin{array}{ll}
 & \bot\!\!\!\bot \;=\; R\circ\top\!\!\!\top\circ S \\
\equiv & \quad \{\text{ excluded middle }\} \\
 & \bot\!\!\!\bot = R\circ\top\!\!\!\top\circ S \;\;\wedge\;\; (R = \bot\!\!\!\bot \vee R \neq \bot\!\!\!\bot) \\
\equiv & \quad \{\text{ predicate calculus, } \bot\!\!\!\bot \text{ is a zero of composition }\} \\
 & R = \bot\!\!\!\bot \;\;\vee\;\; (\bot\!\!\!\bot = R\circ\top\!\!\!\top\circ S \;\;\wedge\;\; R \neq \bot\!\!\!\bot) \\
\equiv & \quad \{\text{ lemma 42 }\} \\
 & R = \bot\!\!\!\bot \;\;\vee\;\; (\bot\!\!\!\bot = \top\!\!\!\top\circ R\circ\top\!\!\!\top\circ S \;\;\wedge\;\; R \neq \bot\!\!\!\bot) \\
\equiv & \quad \{\text{ cone rule and calculus }\} \\
 & \bot\!\!\!\bot = R \;\;\vee\;\; \bot\!\!\!\bot = \top\!\!\!\top\circ S \\
\equiv & \quad \{\text{ lemma 42 }\} \\
 & \bot\!\!\!\bot = R \;\;\vee\;\; \bot\!\!\!\bot = S
\end{array}$$

□

5.2 The Axioms

In order to capture the notion of a unit type we need to express a sort of dual to the cone rule, namely that there is a non-empty spec which, when viewed as a set of pairs, consists of at most one pair the two components of which are identical. Specifically, we posit the existence of a spec, denoted **1**, such that

(44) $\quad \bot\!\!\!\bot \neq \mathbf{1}$

and

(45) $\quad I \sqsupseteq \mathbf{1}\circ\top\!\!\!\top\circ\mathbf{1}$

There are several ways to convince oneself that axioms (44) and (45) are indeed what we seek. One is to interpret the axioms in the relational model; another is to explore the consequences of the axioms within the theory itself. We would not discourage the reader from doing the former, but prefer ourself to emphasise the latter. We verify, first, that the unit type is an "atomic" monotype ("atomic" to be defined shortly) and, second, that it is a "terminal object" in the sense of category theory. Finally, we summarise certain basic properties of the unit type.

5.3 An Atomic Monotype

We begin by verifying that the unit type is a monotype.

Theorem 46 **1** is a monotype.

Proof

$$\begin{array}{ll}
 & I \sqsupseteq \mathbf{1} \\
\equiv & \quad \{\text{ definition of left domain }\} \\
 & I \;\sqsupseteq\; \mathbf{1}_< \circ \mathbf{1} \\
\Leftarrow & \quad \{\; \mathbf{1}\circ\top\!\!\!\top \sqsupseteq \mathbf{1}\circ\mathbf{1}^\cup \sqsupseteq \;\{\text{ definition }\}\; \mathbf{1}_< \;\} \\
 & I \;\sqsupseteq\; \mathbf{1}\circ\top\!\!\!\top\circ\mathbf{1} \\
\equiv & \quad \{\; (45) \;\} \\
 & \textbf{true}
\end{array}$$

□

We now define an *atom* to be a spec R such that, for every spec X,

$$R \sqsupseteq X \ \Rightarrow\ \perp\!\!\perp = X \ \vee\ R = X$$

Clearly, $\perp\!\!\perp$ is an atom. In general, the relational interpretation of an atom is a set of pairs containing at most one element.

Theorem 47 $\quad \mathbf{1}$ is an atom

Proof Let X be a spec such that $\mathbf{1} \sqsupseteq X$. Then, by the definition of an atom, we must prove that $\perp\!\!\perp = X \ \vee\ \mathbf{1} = X$.

$$
\begin{array}{ll}
 & \perp\!\!\perp = X \ \vee\ \mathbf{1} = X \\
\equiv & \quad \{ \mathbf{1} \sqsupseteq X \} \\
 & \perp\!\!\perp = X \ \vee\ \mathbf{1} \sqcap \neg X = \perp\!\!\perp \\
\equiv & \quad \{ \text{lemma 43} \} \\
 & \perp\!\!\perp \ = \ X \circ \top\!\!\top \circ (\mathbf{1} \sqcap \neg X) \\
\equiv & \quad \{ I \sqsupseteq \{(45)\}\ \mathbf{1} \circ \top\!\!\top \circ \mathbf{1} \sqsupseteq X \circ \top\!\!\top \circ (\mathbf{1} \sqcap \neg X), (1) \} \\
 & \perp\!\!\perp \ = \ X \circ \top\!\!\top \circ (\mathbf{1} \sqcap \neg X) \circ X \circ \top\!\!\top \circ (\mathbf{1} \sqcap \neg X) \\
\Leftarrow & \quad \{ \perp\!\!\perp \text{ is a zero of composition} \} \\
 & \perp\!\!\perp \ = \ (\mathbf{1} \sqcap \neg X) \circ X \\
\equiv & \quad \{ I \sqsupseteq \mathbf{1} \sqsupseteq X, (2) \} \\
 & \perp\!\!\perp \ = \ \mathbf{1} \sqcap \neg X \sqcap X \\
\equiv & \quad \{ \text{contradiction} \} \\
 & \mathbf{true}
\end{array}
$$

□

Properties (44), (46) and (47) express, respectively, that **1** is non-empty, and is a monotype corresponding to a set containing at most one element.

5.4 Terminality

The abstractness in the definition of the unit type consists, in part, of the fact that the unit type characterises *any* one-element set (or, if you prefer, is modelled by any one-element set); the identity of the element is irrelevant. In category theory a unit type is characterised by the following so-called "terminality" property: for each set A, there is one, and only one, function — commonly denoted by $!_A$ — in $\mathbf{1} \longleftarrow A$. Letting $!$ denote $\mathbf{1} \circ \top\!\!\top$, this characterisation of the unit type is mimicked in our theory by the following two consequences of axioms (44) and (45). For all monotypes A,

(48) $\quad ! \circ A \in \mathbf{1} \longleftarrow A$

(49) $\quad R \in \mathbf{1} \sim A \ \wedge\ R> \,=\, A \ \equiv\ R \,=\, ! \circ A$

Thus the categorical function $!_A$ is rendered by the imp $! \circ A$.

Equivalent, more succinct, and more fundamental, renderings of (48) and (49) are

(50) $\quad !$ is an imp, and

(51) $\quad \mathbf{1} \ = \ \mathbf{1} \circ \top\!\!\top \circ \mathbf{1}$

from which follows

(52) $\quad \mathbf{1} \sqsupseteq R< \ \equiv\ R \,=\, ! \circ R>$

Here are their proofs.

Proof of (50)

$$
\begin{array}{ll}
 & !\ \ \text{is an imp} \\
\equiv & \quad \{\ \text{definition, reverse}\ \} \\
 & I\ \sqsupseteq\ \mathbf{1} \circ \top\!\!\top \circ \top\!\!\top\cup \circ \mathbf{1}\cup \\
\equiv & \quad \{\ \mathbf{1}\ \text{is a monotype, (1), properties of}\ \top\!\!\top\ \} \\
 & I\ \sqsupseteq\ \mathbf{1} \circ \top\!\!\top \circ \mathbf{1} \\
\equiv & \quad \{\ 45\ \} \\
 & \mathbf{true}
\end{array}
$$

□

Proof of (51)

$$
\begin{array}{ll}
 & \mathbf{1} \circ \top\!\!\top \circ \mathbf{1} \\
\sqsupseteq & \quad \{\ \top\!\!\top \sqsupseteq I\ \} \\
 & \mathbf{1} \circ \mathbf{1} \\
= & \quad \{\ \mathbf{1}\ \text{is a monotype, (1)}\ \} \\
 & \mathbf{1} \\
\sqsupseteq & \quad \{\ \text{(45), monotonicity of composition}\ \} \\
 & \mathbf{1} \circ \mathbf{1} \circ \top\!\!\top \circ \mathbf{1} \\
= & \quad \{\ \mathbf{1}\ \text{is a monotype, (1)}\ \} \\
 & \mathbf{1} \circ \top\!\!\top \circ \mathbf{1}
\end{array}
$$

□

Proof of (52)

$$
\begin{array}{ll}
 & \mathbf{1}\ \sqsupseteq\ R< \\
\equiv & \quad \{\ (5)\ \} \\
 & R\ =\ \mathbf{1} \circ R \\
\equiv & \quad \{\ (51)\ \} \\
 & R = \mathbf{1} \circ \top\!\!\top \circ \mathbf{1} \circ R\ \wedge\ R = \mathbf{1} \circ R \\
\equiv & \quad \{\ (1),\ \mathbf{1}\ \text{is a monotype}\ \} \\
 & R\ =\ \mathbf{1} \circ \top\!\!\top \circ R \\
\equiv & \quad \{\ (14)\ \text{with}\ R := S\ \text{and}\ S := \top\!\!\top\ \} \\
 & R\ =\ \mathbf{1} \circ \top\!\!\top \circ R>
\end{array}
$$

□

It is also clear from these properties that **1** is unique up to isomorphism: if $\mathbf{1}'$ is also a unit type then $\mathbf{1} \circ \top\!\!\top \circ \mathbf{1}'$ is a bijection between **1** and $\mathbf{1}'$.

5.5 A Summary of Basic Properties

The "foundations" that were laid in sections 3 and 4 were not without purpose. In this and later sections we shall continually ask a number of standard questions about the specs and/or operators that have been newly introduced, the questions falling under headings such as "left and right domains", "imps and co-imps", and "natural polymorphism". Two such questions have already been answered for the unit type: it is a monotype and the spec ! is an imp. To these we might also add that the function from specs to specs that always returns **1** is a relator (because **1** is a monotype). This seemingly trivial remark will prove to be quite important. There are two "standard questions" yet to be answered: what are the left and right domains of ! and in what sense is it naturally polymorphic? Here is the answer to the first of these.

Theorem 53

(a) $!< \; = \; \mathbf{1}$
(b) $!> \; = \; I$
□

Verification of both of these is straightforward and is left to the reader. (For part (a) make use of (51). For part (b) make use of the cone rule.)
The final question in this list is answered by the following theorem.

Theorem 54

$$! \in \mathbf{1} \overset{\cdot}{\leftrightsquigarrow} \top\!\!\top$$

In particular, for all specs R,

$$! \in \mathbf{1} \overset{\cdot}{\leftsquigarrow} R$$

Proof

$$\begin{array}{ll}
 & ! \in \mathbf{1} \overset{\cdot}{\leftrightsquigarrow} \top\!\!\top \\
\equiv & \qquad \{ \text{ definition } !, \text{ definition } \in \ \} \\
 & \mathbf{1} \circ \top\!\!\top \ (\mathbf{1} \overset{\cdot}{\leftrightsquigarrow} \top\!\!\top) \ \mathbf{1} \circ \top\!\!\top \\
\equiv & \qquad \{ \text{ definition } \overset{\cdot}{\leftrightsquigarrow} \ \} \\
 & \mathbf{1} \circ \mathbf{1} \circ \top\!\!\top \ = \ \mathbf{1} \circ \top\!\!\top \circ \top\!\!\top \\
\equiv & \qquad \{ \ \mathbf{1} \text{ is a monotype, } \top\!\!\top \ = \ \top\!\!\top \circ \top\!\!\top \ \} \\
 & \mathbf{true}
\end{array}$$

The corollary follows because equality is a special case of inclusion and $\overset{\cdot}{\leftsquigarrow}$ is anti-monotonic in its second argument.
□

The second naturality property of ! above is much the weaker of the two but may have a more familiar appearance. It is derived from the type statement

$$! \circ A \in \mathbf{1} \longleftarrow A$$

by omitting the restriction of the domain to monotype A (in effect considering the polymorphic imp rather than an instance of it), replacing A by an arbitrary spec R and replacing "$\longleftarrow$" by "$\overset{\cdot}{\leftsquigarrow}$". It is this that is often meant by saying that ! is "naturally polymorphic".

The unit type constitutes a building block for the construction of data types; we turn now to the mortar: cartesian product and disjoint sum.

6 Axioms for Cartesian Product and Disjoint Sum

In all systems that we know of, cartesian product and disjoint sum are duals of each other. (Disjoint sum is indeed often given the name "co-product".) In choosing an axiomatisation of the two concepts in a relational framework we have therefore striven for two sets of rules that are "dual" to each other in some clearly recognisable way. It is for this reason that we present the two sets of axioms together in this section. In subsequent sections we consider separately the consequences of the axioms for cartesian product and for disjoint sum before returning in the final section to consider natural isomorphisms between combinations of the two.

In choosing our axioms, we have, of course, been strongly influenced by our experience with set-theoretic presentations of the relational calculus, that being the model our axioms are intended to capture. Since our notation is somewhat unconventional we shall frequently refer to this model for motivation.

We begin by postulating the existence of four specs, for cartesian product the two projections $\ll$ (pronounced "project left") and $\gg$ (pronounced "project right") and for disjoint sum the two injections $\hookrightarrow$ (pronounced "inject left") and $\hookleftarrow$ (pronounced "inject right"). (Note the unconventional direction of the arrow heads. As an aid to memory, and motivation for this choice, we suggest that the reader bear in mind the diagram "$X \hookrightarrow X{+}Y \hookleftarrow Y$".) Further, experience leads us to introduce four binary operators on specs, for cartesian product $\vartriangle$ (pronounced "split") and $\times$ (pronounced "times"), and for disjoint sum $\triangledown$ (pronounced "junc") and $+$ (pronounced "plus"), defined in terms of the projection and injection specs as follows:

$$\begin{array}{llcl}
(55) & P \vartriangle Q & = & (\ll^{\cup} \circ P) \sqcap (\gg^{\cup} \circ Q) \\
(56) & P \triangledown Q & = & (P \circ \hookrightarrow^{\cup}) \sqcup (Q \circ \hookleftarrow^{\cup}) \\
(57) & P \times Q & = & (P \circ \ll) \vartriangle (Q \circ \gg) \\
(58) & P{+}Q & = & (\hookrightarrow \circ P) \triangledown (\hookleftarrow \circ Q)
\end{array}$$

The relational model that we envisage assumes that the universe is a term algebra formed by closing some base set under three operators: the binary operator mapping the pair of terms x, y to the term (x,y), and two unary operators $\hookrightarrow$ and $\hookleftarrow$ mapping the term x to the terms $\hookrightarrow.x$ and $\hookleftarrow.x$, respectively. The interpretation of $\ll$ and $\gg$ is that they project a pair onto its left and right components. That is,

$$\begin{array}{l}
x\,\langle\ll\rangle\,(x,y) \\
y\,\langle\gg\rangle\,(x,y)
\end{array}$$

The four defined operators should be familiar from their interpretations which are

$$\begin{array}{lcl}
(x,y)\,\langle P \vartriangle Q\rangle\, z & \equiv & x\,\langle P\rangle\, z \;\wedge\; y\,\langle Q\rangle\, z \\
x\,\langle P \triangledown Q\rangle\, y & \equiv & \exists(z :: y = \hookrightarrow.z \;\wedge\; x\,\langle P\rangle\, z) \\
 & & \vee\; \exists(z :: y = \hookleftarrow.z \;\wedge\; x\,\langle Q\rangle\, z) \\
(u,v)\,\langle P \times Q\rangle\,(x,y) & \equiv & u\,\langle P\rangle x \;\wedge\; v\,\langle Q\rangle y \\
x\,\langle P{+}Q\rangle\, y & \equiv & \exists(u,v :: x = \hookrightarrow.u \;\wedge\; y = \hookrightarrow.v \;\wedge\; u\,\langle P\rangle\, v) \\
 & & \vee\; \exists(u,v :: x = \hookleftarrow.u \;\wedge\; y = \hookleftarrow.v \;\wedge\; u\,\langle Q\rangle\, v)
\end{array}$$

Note that these are the *definitions* of the operators in higher-order SPEC algebras.

Our first axiom is that the injections are both imps.

$$(59) \qquad I \;\sqsupseteq\; (\hookrightarrow \circ \hookrightarrow^{\cup}) \sqcup (\hookleftarrow \circ \hookleftarrow^{\cup})$$

The "dual" of this axiom that we propose is:

$$(60) \qquad I \;\sqsupseteq\; (\ll^{\cup} \circ \ll) \sqcap (\gg^{\cup} \circ \gg)$$

which says that projecting a pair onto its first and second components and then recombining the components leaves the pair unchanged.

(Berghammer and Zierer [4] and de Roever [15] introduce an almost identical axiom to (60) but in their case the axiom is an equality rather than an inclusion. The difference is that their theories are monomorphic and not polymorphic. Relations are assumed to be (externally) typed and there is a family of product operators indexed by pairs of types. In our theory types (or rather domains) are internal and there is just one (polymorphic) product operator.)

We remark that axioms (59) and (60) take the following form when rephrased in terms of the product and sum operations.

(61) $\quad I \;\sqsupseteq\; I{+}I$

(62) $\quad I \;\sqsupseteq\; I\times I$

This is reassuring since it is one step on the way to guaranteeing that $+$ and $\times$ are binary relators.

Cartesian product and conjunction are closely related. Specifically, we have (in the set-theoretic interpretation of $\times$)

$$x\,\langle P\cap Q\rangle\,y \;\equiv\; (x,x)\,\langle P\times Q\rangle\,(y,y)$$

Abstracting from this property in order to find an axiom that has a pleasing syntactic shape we are led to the following axiom:

(63) $\quad (P \vartriangle Q)^{\cup} \circ (R \vartriangle S) \;=\; (P^{\cup} \circ R)\sqcap(Q^{\cup} \circ S)$

The dual axiom for disjoint sum is:

(64) $\quad (P \triangledown Q) \circ (R \triangledown S)^{\cup} \;=\; (P \circ R^{\cup})\sqcup(Q \circ S^{\cup})$

(The reader may wish to interpret these properties in the relational model to assure themself of their validity.)

As a final axiom we postulate that left projection is possible if and only if right projection is possible:

(65) $\quad \ll> \;=\; \gg>$

Property (65) is equivalent to

(66) $\quad \top\!\!\top \circ \ll \;=\; \top\!\!\top \circ \gg$

Its dual is therefore the trivially true

$$\hookrightarrow \circ \perp\!\!\perp \;=\; \hookleftarrow \circ \perp\!\!\perp$$

There are thus no further axioms for disjoint sum.

The five properties (59), (60), (63), (64) and (65) are the sum total of our axiomatisation of cartesian product and disjoint sum.

In the following two sections we consider individually the consequences of the axioms for cartesian product and disjoint sum. The cap operator in the definition of split together with the fact that composition is not universally cap-junctive make the calculations with cartesian product somewhat harder than those with disjoint sum. For this reason we begin with cartesian product and allow ourselves the luxury of much greater brevity in the discussion of disjoint sum. It should be noted that the organisation of the calculations in the next two sections is intended to facilitate, above all, ease of reference. A consequence thereof is that the reader may spot ways of shortening our calculations by interchanging the order of presentation.

7 Properties of Cartesian Product

There is a major complicating factor in developing a *relational* rather than a *functional* theory of datatypes. It is not, however, a complication that we want to avoid or brush under the carpet since it is an inevitable consequence of the desire to face the issue of nondeterminism. The complication can be pinpointed to cartesian product. Consider, as a first example, the "doubling function", i.e. a function that constructs a pair from a singleton by simply copying its argument. This is the imp $I \vartriangle I$. Now consider the equation:

$$(I \vartriangle I) \circ R \quad = \quad R \vartriangle R$$

and let us interpret R as a *nondeterministic function.* The equation is then clearly invalid since on the left side some nondeterministically calculated value is copied whereas on the right side a pair is constructed by applying R twice; since that calculation is nondeterministic the two elements of the pair may differ. If, however, R is a true function (an imp according to our definition) the equation is valid, as can easily be proved. Clearly the difference lies in the fact that imps distribute backwards over the cap operator whereas that is not the case in general.

The ramifications of the lack of such a distributivity property are manyfold. They can best be observed by comparing the theorems in this section with those in the next. In particular the fusion properties in the subsection 7.1, the computation rules in subsection 7.2 and the terminality property in subsection 7.5 are significantly less tractable than their counterparts for disjoint sum.

Many of the theorems in this section go in pairs: one for left and one for right projection. In all cases we prove just one of the two.

7.1 Fusion Properties

Our first concern is whether or not the product operator $(\times)$ is a relator. According to the definition of a relator there are four conditions that we must verify. The first condition is axiomatically true (see (62)). The second condition, the requirement that cartesian product be monotonic in both its arguments, is clear from its definition (it is a composition of monotonic functions). Also clear from the definition of cartesian product is that the reverse operator distributes over it. I.e.

(67) $\quad (P \times Q)^{\cup} \quad = \quad P^{\cup} \times Q^{\cup}$

It remains to show that composition distributes over cartesian product:

Theorem 68 (Product-Split and Product-Product Fusion)

(a) $\quad (P \times Q) \circ (R \vartriangle S) \quad = \quad (P \circ R) \vartriangle (Q \circ S)$

(b) $\quad (P \times Q) \circ (R \times S) \quad = \quad (P \circ R) \times (Q \circ S)$

Proof We only prove the (a)-part, the other part follows immediately from (a) and (57).

$$\begin{array}{cl} & (P \times Q) \circ (R \vartriangle S) \\ = & \quad \{ \text{ (67), reverse } \} \\ & (P^{\cup} \times Q^{\cup})^{\cup} \circ (R \vartriangle S) \\ = & \quad \{ \text{ (57) } \} \\ & ((P^{\cup} \circ \ll) \vartriangle (Q^{\cup} \circ \gg))^{\cup} \circ (R \vartriangle S) \\ = & \quad \{ \text{ (63) } \} \end{array}$$

$$\begin{array}{ll} & (P^{\cup} \circ \ll)^{\cup} \circ R \ \sqcap \ (Q^{\cup} \circ \gg)^{\cup} \circ S \\ = & \quad \{ \text{ reverse } \} \\ & \ll^{\cup} \circ \ P \circ R \ \ \sqcap \ \gg^{\cup} \circ \ Q \circ S \\ = & \quad \{ \ (55) \ \} \\ & (P \circ R) \ \vartriangle \ (Q \circ S) \end{array}$$

□

Properties (68a) and (68b) are the first examples of many properties to which we give the name "fusion" property. In general, whenever we introduce a relator we seek its associated "morphism" operator (in the case of cartesian product this is split, and in the case of disjoint sum this is junc) and we investigate conditions under which two specs can be "fused" into the one morphism. (Typically, as in (68a) and (68b) one of the specs to be fused is itself a morphism.) Elsewhere [3] we discuss relators defined via fixed-points and observe a connection between morphism fusion and loop fusion. Note, however, that we do not always use the rules to "fuse" specs; just as often we use them to "defuse" a spec into component parts. The reader should not allow the one-way character of the name to prejudice their use of such rules.

Remark Our efforts to identify categories of properties to which we give compact names can never be wholly satisfactory because the categories are not distinct. Property 68(b), for instance, is both a fusion property — because a product is a particular form of morphism — and an abide law — composition and product abide with each other. *End of Remark*

Corollary 69 $\times$ is a binary relator.

□

A fusion equality in which the split occurs to the left of the composition cannot be established in general. An inclusion does hold, however, and is not entirely useless. Two cases where an equality can be established (although not the only ones) are when one operand of the split has the form $S \circ \top\!\top$ for some S and when the right operand of the composition is an imp.

Theorem 70 (Split-Imp Fusion)

(a) $(R \vartriangle S) \circ T \ \sqsubseteq \ (R \circ T) \ \vartriangle \ (S \circ T)$

(b) $(R \vartriangle (S \circ \top\!\top)) \circ T \ = \ (R \circ T) \ \vartriangle \ (S \circ \top\!\top)$

and, for all imps f,

(b) $(R \vartriangle S) \circ f \ = \ (R \circ f) \ \vartriangle \ (S \circ f)$

Proof Straightforward unfolding of the definition of split augmented by, in the case of part (a), monotonicity, in the case of part (b), (19), and in the case of part (c), the cap-distributivity property of imps.

□

7.2 Computation Rules

The name "projection" immediately suggests its operational interpretation. Here that interpretation is represented by two rules that we call "computation rules". Before we can derive these rules we need to note several lemmas:

Lemma 71

(a) $\top\!\top \circ \gg \ \sqsupseteq \ \ll$

(b) $\top\!\top \circ \ll \ \sqsupseteq \ \gg$

Proof Immediate from (66), $\top\!\!\top \circ \ll \; \sqsupseteq \; \ll$ and $\top\!\!\top \circ \gg \; \sqsupseteq \; \gg$.

□

We shall have further use of lemma 71 later, but an immediate corollary is that one can express various combinations of one projection in terms of the split and product operators:

Lemma 72

$$\begin{array}{lrcl}
\text{(a)} & \ll^{\cup} & = & I \vartriangle \top\!\!\top \\
\text{(b)} & \gg^{\cup} & = & \top\!\!\top \vartriangle I \\
\text{(c)} & \ll^{\cup} \circ \ll & = & I \times \top\!\!\top \\
\text{(d)} & \gg^{\cup} \circ \gg & = & \top\!\!\top \times I
\end{array}$$

Proof We only prove (a).

$$\begin{array}{ll}
 & I \vartriangle \top\!\!\top \\
= & \quad \{ (55) \} \\
 & \ll^{\cup} \circ I \;\sqcap\; \gg^{\cup} \circ \top\!\!\top \\
= & \quad \{ \text{lemma 71, reverse} \} \\
 & \ll^{\cup}
\end{array}$$

To prove (c) use the same two properties plus properties of reverse.

□

The following theorem is the announced "computation rule" permitting the "execution" (or simplification) of a projection. Note that the rule is valid for all specs P and Q but the righthand side of each rule is slightly more complex than a naive examination might suggest.

Theorem 73 (Computation Rules for Split)

$$\begin{array}{lrcl}
\text{(a)} & \ll \circ (P \vartriangle Q) & = & P \circ Q{>} \\
\text{(b)} & \gg \circ (P \vartriangle Q) & = & Q \circ P{>}
\end{array}$$

Proof

$$\begin{array}{ll}
 & \ll \circ (P \vartriangle Q) \\
= & \quad \{ (72), \text{reverse} \} \\
 & (I \vartriangle \top\!\!\top)^{\cup} \circ (P \vartriangle Q) \\
= & \quad \{ (63) \} \\
 & P \;\sqcap\; \top\!\!\top \circ Q \\
= & \quad \{ (14) \} \\
 & P \circ Q{>}
\end{array}$$

□

We mention one, easily derived, corollary of lemma 72 and theorem 73.

Theorem 74

$$\ll \circ \gg^{\cup} \;=\; \top\!\!\top \;=\; \gg \circ \ll^{\cup}$$

□

Theorem 74 is important if only because it is an important stepping stone to proving that product is "strict".

Theorem 75 (Product is Strict)

$$R \times S \;=\; \perp\!\!\perp \quad\equiv\quad R \;=\; \perp\!\!\perp \;\vee\; S \;=\; \perp\!\!\perp$$

Proof

$$\begin{array}{ll}
 & R \times S \;=\; \perp\!\!\perp \\
\equiv & \quad \{ \text{ definition of product, } X \sqcap Y \;=\; \perp\!\!\perp \;\;\equiv\;\; X \circ Y^{\cup} \sqsubseteq \neg I \;\} \\
 & \ll^{\cup} \circ R \circ \ll \circ \gg^{\cup} \circ S^{\cup} \circ \gg \;\;\sqsubseteq\;\; \neg I \\
\equiv & \quad \{ \text{ theorem 74 } \} \\
 & \ll^{\cup} \circ R \circ \top\!\!\top \circ S^{\cup} \circ \gg \;\;\sqsubseteq\;\; \neg I \\
\equiv & \quad \{ \text{ RS rule, properties of reverse } \} \\
 & R^{\cup} \circ \ll \circ I \circ \gg^{\cup} \circ S \;\;\sqsubseteq\;\; \perp\!\!\perp \\
\equiv & \quad \{ \text{ plat calculus, theorem 74 } \} \\
 & R^{\cup} \circ \top\!\!\top \circ S \;\;=\;\; \perp\!\!\perp \\
\equiv & \quad \{ \text{ lemma 43, property of reverse } \} \\
 & R \;=\; \perp\!\!\perp \;\;\vee\;\; S \;=\; \perp\!\!\perp
\end{array}$$

□

The strictness of product offers the first hint as to why one chooses to denote it by the symbol for multiplication; the analogy that is made is with multiplication by zero. Its proof is, in addition, particularly delightful and should not be rushed past. The goal in the calculation is to exploit lemma 43. Since the lemma is symmetric in R and S it would be very disappointing if that symmetry were lost in the first few steps of the calculation. It is this that motivates beginning as we did by using the property

$$X \sqcap Y \;=\; \perp\!\!\perp \quad\equiv\quad X \circ Y^{\cup} \sqsubseteq \neg I$$

to replace the cap operator by composition. Admittedly the latter is a property that has not been mentioned before and one that emerges somewhat out of the blue. However, it is easily proved using the rotation rule and is a property that would certainly be documented in a systematic development of the calculus of relations.

Since product is defined in terms of split, the above computation rules can be instantiated with that definition giving computation rules for product. Doing so, however, one obtains ugly domain expressions that we do not care to use. The next lemma reformulates those expressions and happens to come in handy in a later calculation.

Lemma 76

$$\begin{array}{llllll}
\text{(a)} & (P \circ \ll){>} & = & \ll^{\cup} \circ P{>} \circ \ll \;\sqcap\; \gg^{\cup} \circ \gg & = & P{>} \times I \\
\text{(b)} & (Q \circ \gg){>} & = & \ll^{\cup} \circ \ll \;\sqcap\; \gg^{\cup} \circ Q{>} \circ \gg & = & I \times Q{>}
\end{array}$$

Proof We prove (a) only. Within (a), the equality between the second and third expressions is a straightforward unfolding of the definition of product so we limit our attention to the equality between the first and second expressions.

$$\begin{array}{ll}
 & (P \circ \ll){>} \\
= & \quad \{ \;(15)\; \} \\
 & (P{>} \;\circ\; \ll){>} \\
= & \quad \{ \text{ definition of right domain, } P{>} \text{ is a monotype } \} \\
 & I \;\;\sqcap\;\; \ll^{\cup} \circ P{>} \circ \ll
\end{array}$$

$$\begin{array}{ll}
= & \quad \{\ P{>} \text{ is a monotype, monotonicity }\} \\
& I \sqcap \ll^{\cup} \circ P{>} \circ \ll \ \sqcap \ \ll^{\cup} \circ \ll \\
= & \quad \{\ I \sqcap \ll^{\cup} \circ \ll \ = \ \{(65)\}\ I \sqcap \gg^{\cup} \circ \gg \ \} \\
& I \sqcap \ll^{\cup} \circ P{>} \circ \ll \ \sqcap \ \gg^{\cup} \circ \gg \\
= & \quad \{\ (63), \text{ monotonicity }\} \\
& \ll^{\cup} \circ P{>} \circ \ll \ \sqcap \ \gg^{\cup} \circ \gg
\end{array}$$

□

Theorem 77 (Computation Rules for Product)

$$\begin{array}{llll}
\text{(a)} & \ll \circ (P \times Q) & = & P \circ \ll \circ (I \times Q{>}) \\
\text{(b)} & \gg \circ (P \times Q) & = & Q \circ \gg \circ (P{>} \times I)
\end{array}$$

Proof

$$\begin{array}{ll}
& \ll \circ (P \times Q) \\
= & \quad \{ \text{ definition of } \times, \text{ computation rule 73(a) } \} \\
& P \circ \ll \circ (Q \circ \gg){>} \\
= & \quad \{ (76) \} \\
& P \circ \ll \circ (I \times Q{>})
\end{array}$$

□

7.3 Imp and Co-imp Preservation

Up till now our language has implied that the projections are imps but, as yet, we have not stated the fact so explicitly and nor has it been proven. Not surprisingly the proof is rather trivial.

Lemma 78

$$\ll \circ \ll^{\cup} \ = \ I \quad \text{and} \quad \gg \circ \gg^{\cup} \ = \ I$$

Proof As always we content ourselves with the proof of just one of the statements.

$$\begin{array}{ll}
& \ll \circ \ll^{\cup} \\
= & \quad \{ (72) \} \\
& \ll \circ (I \vartriangle \top\!\!\top) \\
= & \quad \{ \text{ computation rule, (73) } \} \\
& I \circ \top\!\!\top{>} \\
= & \quad \{\ \top\!\!\top{>} = I\ \} \\
& I
\end{array}$$

□

Corollary 79 $\ll$ and $\gg$ are both imps.

Proof Immediate from the definition of an imp.
□

Theorem 80

(a) $P \vartriangle Q$ is a co-imp $\Leftarrow$ P is a co-imp $\vee$ Q is a co-imp.
(b) $P \vartriangle Q$ is an imp $\Leftarrow$ P is an imp $\wedge$ Q is an imp.

Proof Note the disjunction in part (a). This quite strong theorem is nevertheless straightforward to prove by application of axiom (63). Part (b) is a little less straightforward:

$$\begin{array}{ll}
 & R \vartriangle S \text{ is an imp} \\
\equiv & \quad \{ \text{ definition } \} \\
 & I \sqsupseteq (R \vartriangle S) \circ (R \vartriangle S)^{\cup} \\
\equiv & \quad \{ \text{ (55), reverse } \} \\
 & I \sqsupseteq (\ll^{\cup} \circ R \sqcap \gg^{\cup} \circ S) \circ (R^{\cup} \circ \ll \sqcap S^{\cup} \circ \gg) \\
\Leftarrow & \quad \{ \text{ monotonicity } \} \\
 & I \sqsupseteq \ll^{\cup} \circ R \circ R^{\cup} \circ \ll \;\sqcap\; \gg^{\cup} \circ S \circ S^{\cup} \circ \gg \\
\Leftarrow & \quad \{ \text{ (60) } \} \\
 & I \sqsupseteq R \circ R^{\cup} \;\wedge\; I \sqsupseteq S \circ S^{\cup} \\
\equiv & \quad \{ \text{ definition } \} \\
 & R \text{ is an imp } \wedge\; S \text{ is an imp}
\end{array}$$

□

Since product is a binary relator we have:

Theorem 81 $\times$ preserves both imps and co-imps.
□

(The fact that a split is a co-imp if just one of its arguments is a co-imp does not help one to prove anything stronger about product.)

7.4 Left and Right Domains

Much of the work necessary to determine the effect of the left and right domain operators on splits and left and right projections has already been completed. Lemma 78, for instance, tells us immediately that the projections are surjective. I.e.

Theorem 82 $\ll< \; = I$ and $\gg< \; = I$
□

Moreover, lemma 76, with P and Q both instantiated to I, predicts their right domains:

Theorem 83 $\ll> \; = I \times I$ and $\gg> \; = I \times I$
□

Since product is a (binary) relator we can immediately instantiate theorem 31 obtaining:

Theorem 84

(a) $(P \times Q)< \; = \; P< \times Q<$
(b) $(P \times Q)> \; = \; P> \times Q>$
□

We conclude this section with expressions for the right and left domains of a split. That for the left domain is not particularly helpful but is more compact than the expanded form of the definition! (As one might expect it is usually more difficult to predict the left domain than the right domain of a split.)

Theorem 85 (Split Right and Left Domain)

$$\begin{array}{lrcl} \text{(a)} & (P \vartriangle Q)> & = & P> \sqcap Q> \\ \text{(b)} & (P \vartriangle Q)< & = & \ll^{\cup} \circ P \circ Q^{\cup} \circ \gg \sqcap I \end{array}$$

Proof Simple application of the definition of split and right domain together with axiom (63) for part (a) and the definition of split together with property (12) for part (b).
□

7.5 Unique Extension Properties

In the category **Set** of sets and total functions cartesian product is defined via limits of functors in the following way. Let **2** be the discrete category with objects {0,1}. For object A in **Set** the constant A functor is denoted by $\underline{A} : \mathbf{Set} \longleftarrow \mathbf{2}$. The cartesian product of X and Y is defined to be the limit of the functor $\mathcal{F} : \mathbf{Set} \longleftarrow \mathbf{2}$ with $\mathcal{F} \cdot 0 = X$ and $\mathcal{F} \cdot 1 = Y$. I.e. the product is a set C and a natural transformation $\pi : \mathcal{F} \overset{\cdot}{\longleftarrow} \underline{C}$ such that for every set D and every natural transformation $\sigma : \mathcal{F} \overset{\cdot}{\longleftarrow} \underline{D}$ there is a unique arrow $\phi : C \longleftarrow D$ in **Set** such that $\pi \circ \phi = \sigma$. Usually C is denoted by $X \prod Y$ or $X \times Y$, while the natural transformation is denoted by the pair π_X , π_Y of projections. The terminality of π is most often phrased as follows. For every D and all total functions $f : X \longleftarrow D$ and $g : Y \longleftarrow D$ there is a unique total function $h : X \prod Y \longleftarrow D$ such that $\pi_X \circ h = f$ and $\pi_Y \circ h = g$. In our system this terminality is valid not only for imps (our equivalent of functions) but also for a more general class of specs (although not for all specs). We refer to the relevant theorem as the "unique extension property" for cartesian product, and it is the purpose of this section to present the property and then to explore some of its consequences. First, a useful lemma.

Lemma 86

$$P \vartriangle Q \;=\; (P \circ Q>) \vartriangle (Q \circ P>)$$

Proof

$$\begin{array}{ll} & (P \circ Q>) \vartriangle (Q \circ P>) \\ = & \quad \{ \text{ domains (6), monotypes (2) } \} \\ & (P \circ (P> \sqcap Q>)) \vartriangle (Q \circ (P> \sqcap Q>)) \\ = & \quad \{ \; P> \sqcap Q> \text{ is an imp, cap-distributivity } \} \\ & (P \vartriangle Q) \circ (P> \sqcap Q>) \\ = & \quad \{ \text{ (85(a)), (6) } \} \\ & P \vartriangle Q \end{array}$$

□

In its most general form the unique extension property is as follows:

Theorem 87 (Unique Extension Property)
Suppose $\unlhd \in \{\sqsubseteq, =, \sqsupseteq\}$. Assume also that

$$X \;\unlhd\; \ll^{\cup} \circ \ll \circ X \;\sqcap\; \gg^{\cup} \circ \gg \circ X$$

Then

$$X \unlhd P \vartriangle Q \;\equiv\; \ll \circ X \unlhd P \circ Q> \;\wedge\; \gg \circ X \unlhd Q \circ P>$$

Proof The $\Rightarrow$-part of the equivalence follows from the computation rule, theorem 73 and monotonicity. For the other part we assume the righthand side of the equivalence and prove the validity of the lefthand side:

$$\begin{array}{ll}
 & P \vartriangle Q \\
= & \quad \{ \text{ lemma 86) } \} \\
 & (P \circ Q{>}) \vartriangle (Q \circ P{>}) \\
= & \quad \{ \text{ definition of split } \} \\
 & \ll^{\cup} \circ (P \circ Q{>}) \ \sqcap \ \gg^{\cup} \circ (Q \circ P{>}) \\
\trianglerighteq & \quad \{ \text{ rhs is assumed true, monotonicity } \} \\
 & \ll^{\cup} \circ \ll \circ X \ \sqcap \ \gg^{\cup} \circ \gg \circ X \\
\trianglerighteq & \quad \{ \text{ assumption } \} \\
 & X
\end{array}$$

□

More often than not we apply the theorem with the variable "$\trianglelefteq$" instantiated to "$=$". However, since our purpose is to develop a theory that admits program refinement as a possible step we are continually on the lookout for more general properties of the same nature as theorem 87, the cost in terms of burden of proof being typically almost negligible.

The assumption in theorem 87 is somewhat unwieldy; however, it is important to note that it is *not* equivalent to X being an imp but it is only implied by that circumstance (for all instantiations of $\trianglelefteq$). The assumption is indeed quite weak and we shall encounter several instances where it is valid. One such case is where X is itself a split term, resulting in the following cancellation property.

Theorem 88 (Split Cancellation)

For all $\trianglelefteq \in \{\sqsubseteq, =, \sqsupseteq\}$

$$P \vartriangle Q \trianglelefteq R \vartriangle S \ \equiv \ P \circ Q{>} \trianglelefteq R \circ S{>} \ \wedge \ Q \circ P{>} \trianglelefteq S \circ R{>}$$

Proof We aim to use the unique extension property with X instantiated to $R \vartriangle S$. We must therefore verify the premise. Now,

$$\begin{array}{ll}
 & \ll^{\cup} \circ \ll \circ (R \vartriangle S) \ \sqcap \ \gg^{\cup} \circ \gg \circ (R \vartriangle S) \\
= & \quad \{ \text{ computation rule for split } \} \\
 & \ll^{\cup} \circ R \circ S{>} \ \sqcap \ \gg^{\cup} \circ S \circ R{>} \\
= & \quad \{ \text{ definition of split } \} \\
 & (R \circ S{>}) \vartriangle (S \circ R{>}) \\
= & \quad \{ \text{ lemma 86 } \} \\
 & R \vartriangle S
\end{array}$$

Thus the premise is verified (whatever the value of $\trianglelefteq$ since equality implies inclusion). The rest is straightforward: use the unique extension property and then the computation rule for split to eliminate the projections.

□

We return now to the original concern, which was the case that X, P and Q are all imps. The backwards distribution of imps over intersection shows that the assumption in the statement of the unique extension property is met for imps with left domain in $I \times I$. For the terminality we also have to get rid of the right domains. This explains the assumptions in the terminality theorem.

Theorem 89 (Terminality)
Let f be an imp with $I \times I \sqsupseteq f<$ and let $P> \; = \; Q>$. Then

$$f \;=\; P \vartriangle Q \;\;\equiv\;\; \ll \circ f \;=\; P \;\wedge\; \gg \circ f \;=\; Q$$

Equivalently, for all imps f and all specs P, Q,

$$(I \times I) \circ f \;=\; P' \vartriangle Q' \;\;\equiv\;\; \ll \circ f \;=\; P' \;\wedge\; \gg \circ f \;=\; Q'$$

where P' denotes $P \circ Q>$ and Q' denotes $Q \circ P>$.

Proof See the discussion above.
□

7.6 Naturality Properties

Part (a) of lemma 68 is a very important property, just as important as part (b). It can be expressed somewhat differently, namely as a naturality property of split.

Theorem 90 (Naturality of Split)

$$\vartriangle \;\in\; (R \times S \mathrel{\dot{\leftsquigarrow}} T) \mathrel{\dot{\leftsquigarrow}} (R \mathrel{\dot{\leftsquigarrow}} T) \times (S \mathrel{\dot{\leftsquigarrow}} T)$$

Proof

$$\begin{array}{ll}
 & \vartriangle \;\in\; (R \times S \mathrel{\dot{\leftsquigarrow}} T) \mathrel{\dot{\leftsquigarrow}} (R \mathrel{\dot{\leftsquigarrow}} T) \times (S \mathrel{\dot{\leftsquigarrow}} T) \\
\equiv & \qquad \{ \;(38),\;\; \vartriangle \text{ is a function } \} \\
 & \forall(U,V :: \;\; \vartriangle.U \,\langle R \times S \mathrel{\dot{\leftsquigarrow}} T \rangle \vartriangle.V \\
 & \qquad \Leftarrow \;\; U \,\langle (R \mathrel{\dot{\leftsquigarrow}} T) \times (S \mathrel{\dot{\leftsquigarrow}} T) \rangle\, V \\
 &) \\
\equiv & \qquad \{ \text{ definition of } \times \text{ in a higher-order algebra } \} \\
 & \forall(U1,U2,V1,V2 :: \\
 & \qquad\qquad U1 \vartriangle U2 \,\langle R \times S \mathrel{\dot{\leftsquigarrow}} T \rangle\, V1 \vartriangle V2 \\
 & \qquad \Leftarrow \;\; U1 \,\langle R \mathrel{\dot{\leftsquigarrow}} T \rangle\, V1 \;\wedge\; U2 \,\langle S \mathrel{\dot{\leftsquigarrow}} T \rangle\, V2 \\
 &) \\
\equiv & \qquad \{ \text{ definition of } \mathrel{\dot{\leftsquigarrow}} \} \\
 & \forall(U1,U2,V1,V2 :: \\
 & \qquad\qquad (R \times S) \circ (V1 \vartriangle V2) \;\sqsupseteq\; (U1 \vartriangle U2) \circ T \\
 & \qquad \Leftarrow \;\; R \circ V1 \sqsupseteq U1 \circ T \;\wedge\; S \circ V2 \sqsupseteq U2 \circ T \\
 &) \\
\Leftarrow & \qquad \{ \text{ theorem 68(a); split-spec fusion theorem 70(a) } \} \\
 & \forall(U1,U2,V1,V2 :: \\
 & \qquad\qquad (R \circ V1) \vartriangle (S \circ V2) \;\sqsupseteq\; (U1 \circ T) \vartriangle (U2 \circ T) \\
 & \qquad \Leftarrow \;\; R \circ V1 \sqsupseteq U1 \circ T \;\wedge\; S \circ V2 \sqsupseteq U2 \circ T \\
 &) \\
\equiv & \qquad \{ \text{ monotonicity of } \vartriangle \} \\
 & \textbf{true}
\end{array}$$

□

Note the occurrence of "$\Leftarrow$" in the fourth step; it is not the case that any of the "$\mathrel{\dot{\leftsquigarrow}}$" operators can be replaced by either "$\mathrel{\dot{\rightsquigarrow}}$"or "$\mathrel{\dot{\leftrightsquigarrow}}$".

Since product is a (binary) relator we can simply instantiate theorem 39 to obtain:

Theorem 91 (Naturality of Product)

For all specs R, S, T, U and all $\sim\ \in \{\leftsquigarrow, \rightsquigarrow, \leftrightsquigarrow\}$

$$\times \in (R \times T \sim S \times U) \leftsquigarrow (R \sim S) \times (T \sim U)$$

□

The two projections are also naturally polymorphic in the following sense.

Theorem 92 (Naturality of Left and Right Projection)

For all specs R,

(a) $\quad \ll\ \in R \leftrightsquigarrow R \times \top\!\!\top \quad$ and $\quad \gg\ \in R \leftrightsquigarrow \top\!\!\top \times R$

In particular, for all specs R and S,

(b) $\quad \ll\ \in R \leftsquigarrow R \times S \quad$ and $\quad \gg\ \in R \leftsquigarrow S \times R$

(Note that "$\leftrightsquigarrow$" in the statement of the theorem can be replaced by "$\leftsquigarrow$" or "$\rightsquigarrow$" since equality implies inclusion.)

Proof

$$\begin{array}{ll}
 & \ll\ \in R \leftrightsquigarrow R \times \top\!\!\top \\
\equiv & \qquad \{ \text{ definition of } \leftrightsquigarrow \ \} \\
 & R \circ \ll\ =\ \ll \circ (R \times \top\!\!\top) \\
\equiv & \qquad \{ \text{ computation rule (77) } \} \\
 & R \circ \ll\ =\ R \circ \ll \circ (I \times \top\!\!\top>) \\
\equiv & \qquad \{ \ \top\!\!\top> \ = I, \text{ theorem 83 } \} \\
 & \mathbf{true}
\end{array}$$

The corollary follows because $\leftsquigarrow$ is antimonotonic in its right argument (as is easily verified).
□

Now that we have cartesian product we can make the statement of the polymorphic type of composition more compact.

Theorem 93 (Naturality of Composition)
For all $\sim\ \in \{\leftsquigarrow, \rightsquigarrow, \leftrightsquigarrow\}$, and all specs P, Q and R,

$$\circ \in (P \sim R) \leftsquigarrow (P \sim Q) \times (Q \sim R)$$

□

7.7 Junctivity Properties

Although we give more general junctivity properties below, we start with the finite junctivities. Distribution of cap over split and product is given by

Theorem 94 (Split-Cap and Product-Cap Abide Laws)

(a) $\quad (P \vartriangle Q) \sqcap (R \vartriangle S) \ = \ (P \sqcap R) \vartriangle (Q \sqcap S)$
(b) $\quad (P \times Q) \sqcap (R \times S) \ = \ (P \sqcap R) \times (Q \sqcap S)$

Proof We only prove (a). The proof of (b) is almost the same, but for an extra appeal to the backward distribution of composition over imps.

$$
\begin{array}{ll}
& (P \vartriangle Q) \sqcap (R \vartriangle S) \\
= & \quad \{ (55)\ ;\ \text{plat calculus} \} \\
& \ll^{\cup} \circ\, P \;\sqcap\; \ll^{\cup} \circ\, R \;\sqcap\; \gg^{\cup} \circ\, Q \;\sqcap\; \gg^{\cup} \circ\, S \\
= & \quad \{ \ll^{\cup} \text{ and } \gg^{\cup} \text{ are co-imps} \} \\
& \ll^{\cup} \circ\, (P \sqcap R) \;\sqcap\; \gg^{\cup} \circ\, (Q \sqcap S) \\
= & \quad \{ (55) \} \\
& (P \sqcap R) \vartriangle (Q \sqcap S)
\end{array}
$$

□

Distribution of cup over split and product has another form. The easy proof is left to the reader.

Theorem 95

(a) $(P \sqcup Q) \vartriangle (R \sqcup S) \;=\; P \vartriangle R \,\sqcup\, P \vartriangle S \,\sqcup\, Q \vartriangle R \,\sqcup\, Q \vartriangle S$

(b) $(P \sqcup Q) \times (R \sqcup S) \;=\; P \times R \,\sqcup\, P \times S \,\sqcup\, Q \times R \,\sqcup\, Q \times S$

In particular,

(c) $Q \vartriangle (R \sqcup S) \;=\; Q \vartriangle R \,\sqcup\, Q \vartriangle S$

(d) $(P \sqcup Q) \vartriangle R \;=\; P \vartriangle R \,\sqcup\, Q \vartriangle R$

(e) $Q \times (R \sqcup S) \;=\; Q \times R \,\sqcup\, Q \times S$

(f) $(P \sqcup Q) \times R \;=\; P \times R \,\sqcup\, Q \times R$

□

As mentioned we can do a lot better than (94) and (95): the split and product operators are positively cap-junctive. They are *not* universally cap-junctive, for in general

$$\top\!\!\top \vartriangle \top\!\!\top \;\neq\; \top\!\!\top \qquad \text{and} \qquad \top\!\!\top \times \top\!\!\top \;\neq\; \top\!\!\top$$

(As a matter of fact the second of the above is independent of our axioms.)

Theorem 96 Let $\mathcal{V}$ be a non-empty bag of pairs (V_λ, V_ρ) of specs, $L = \sqcap(V : V \in \mathcal{V} : V_\lambda)$ and $R = \sqcap(V : V \in \mathcal{V} : V_\rho)$. Then

(a) $L \vartriangle R \;=\; \sqcap(V : V \in \mathcal{V} : V_\lambda \vartriangle V_\rho)$

(b) $L \times R \;=\; \sqcap(V : V \in \mathcal{V} : V_\lambda \times V_\rho)$

Proof

$$
\begin{array}{ll}
& \sqcap(V : V \in \mathcal{V} :\, \ll^{\cup} \circ\, V_\lambda \;\sqcap\; \gg^{\cup} \circ\, V_\rho) \\
= & \quad \{ \text{quantifier calculus} \} \\
& \sqcap(V : V \in \mathcal{V} :\, \ll^{\cup} \circ\, V_\lambda) \sqcap \sqcap(V : V \in \mathcal{V} :\, \gg^{\cup} \circ\, V_\rho) \\
= & \quad \{ \ll \text{ and } \gg \text{ are imps and } \mathcal{V} \text{ is non-empty} \} \\
& \ll^{\cup} \circ\, L \;\sqcap\; \gg^{\cup} \circ\, R \\
= & \quad \{ (55) \} \\
& L \vartriangle R
\end{array}
$$

The proof of part (b) is similar, thus left to the reader.

□

In particular, split and product are cap-continuous. Although they are not cup-junctive, they are cup-continuous:

Theorem 97 Let $\mathcal{V}$ be a linear bag of pairs (V_λ, V_ρ) of specs, $L = \sqcup(V : V \in \mathcal{V} : V_\lambda)$ and $R = \sqcup(V : V \in \mathcal{V} : V_\rho)$. Then

$$\begin{array}{lrcl} \text{(a)} & L \vartriangle R & = & \sqcup(V : V \in \mathcal{V} : V_\lambda \vartriangle V_\rho) \\ \text{(b)} & L \times R & = & \sqcup(V : V \in \mathcal{V} : V_\lambda \times V_\rho) \end{array}$$

Proof

$$\begin{array}{ll} & \ll^{\cup} \circ L \sqcap \gg^{\cup} \circ R \\ = & \quad \{ \text{ universal cup-junctivity of composition } \} \\ & \sqcup(V : V \in \mathcal{V} : \ll^{\cup} \circ V_\lambda) \sqcap \sqcup(V : V \in \mathcal{V} : \gg^{\cup} \circ V_\rho) \\ = & \quad \{ \text{ quantifier calculus } \} \\ & \sqcup(V, W : V, W \in \mathcal{V} : \ll^{\cup} \circ V_\lambda \sqcap \gg^{\cup} \circ W_\rho) \\ = & \quad \{ \mathcal{V} \text{ is linear, diagonalization } \} \\ & \sqcup(V : V \in \mathcal{V} : \ll^{\cup} \circ V_\lambda \sqcap \gg^{\cup} \circ V_\rho) \\ = & \quad \{ (55) \} \\ & \sqcup(V : V \in \mathcal{V} : V_\lambda \vartriangle V_\rho) \end{array}$$

□

8 Properties of Disjoint Sum

We have discussed the properties of cartesian product before those of disjoint sum because the latter are substantially simpler to derive. This is because the cap operator in the definition of split is replaced by the cup operator in the definition of junc, and composition is universally cup-junctive but not universally cap-junctive. Calculations with split and/or the projections can thus often be transliterated into calculations with junc and/or the injections — but less often the other way round. We shall take advantage of this fact by simply stating several properties of disjoint sum without accompanying proof. Only where the claimed property is stronger than its counterpart do we provide a proof. The order of presentation also remains the same so that the reader may compare the properties one-by-one. (Note that we said that proofs about cartesian product can *often* be transliterated. We do not know of an algorithm to perform the transliteration and we shall indeed encounter one example where we were unable to derive a proof by such means. The reader should therefore be on their guard as we are on ours.)

8.1 Fusion Properties

As was the case for cartesian product it is straightforward to see that $+$ satisfies three of the conditions necessary for it to be a relator: the first is satisfied axiomatically, and monotonicity and commutation with reverse are satisfied by construction. Distributivity with respect to composition is also a special case of a "fusion" law, namely that a sum can be fused with a junc.

Theorem 98 (Junc-Sum and Sum-Sum Fusion)

$$\begin{array}{lrcl} \text{(a)} & (P \triangledown Q) \circ (R+S) & = & (P \circ R) \triangledown (Q \circ S) \\ \text{(b)} & (P+Q) \circ (R+S) & = & (P \circ R) + (Q \circ S) \end{array}$$

□

Proof Transliteration of the proof of theorem 68.

□

Corollary 99 $+$ is a relator.

□

One more fusion property can be added to this list on account of the universal cup-junctivity of composition, namely:

Theorem 100 (Spec-Junc Fusion)

$$P \circ (Q \triangledown R) \;=\; (P \circ Q) \triangledown (P \circ R)$$

□

Comparison should be made with theorem 70 where a restriction to imps had to be made in order to obtain an equality.

8.2 Computation Rules

The computation rules for junc do not involve any extra complications (unlike those for split). Their derivation, however, follows the same pattern. Lemma 71 has a trivial counterpart; the following is the counterpart of lemma 72.

Lemma 101

(a) $\hookrightarrow^{\cup} \;=\; I \triangledown \perp\!\!\!\perp$
(b) $\hookleftarrow^{\cup} \;=\; \perp\!\!\!\perp \triangledown I$
(c) $\hookrightarrow \circ \hookrightarrow^{\cup} \;=\; I + \perp\!\!\!\perp$
(d) $\hookleftarrow \circ \hookleftarrow^{\cup} \;=\; \perp\!\!\!\perp + I$

□

For want of inventiveness we give the name "co-strictness" to the dual of the strictness of product.

Theorem 102 (Co-strictness of Sum)

$$R{+}S = \perp\!\!\!\perp \;\equiv\; R = \perp\!\!\!\perp \,\wedge\, S = \perp\!\!\!\perp$$

□

The proof is elementary.

Derivation of the computation rules is now straightforward and is left to the reader.

Theorem 103 (Computation Rules for Junc and Sum)

(a) $(P \triangledown Q) \circ \hookrightarrow \;=\; P$
(b) $(P \triangledown Q) \circ \hookleftarrow \;=\; Q$
In particular
(c) $(P{+}Q) \circ \hookrightarrow \;=\; \hookrightarrow \circ P$
(d) $(P{+}Q) \circ \hookleftarrow \;=\; \hookleftarrow \circ Q$

□

As for split, we mention one particularly interesting corollary obtained by combining the computation rules with lemma 101.

Theorem 104

$$\hookrightarrow^{\cup} \circ \hookleftarrow \;=\; \perp\!\!\!\perp \;=\; \hookleftarrow^{\cup} \circ \hookrightarrow$$

□

8.3 Imp and Co-imp Preservation

Our first axiom was that left and right injection are both imps. In fact they are also co-imps as is evidenced by the following:

Lemma 105

$$\hookrightarrow^{\cup} \circ \hookrightarrow \quad = \quad I \quad = \quad \hookleftarrow^{\cup} \circ \hookleftarrow$$

Proof Immediate from the computation rule (103) combined with (101).
□

Corollary 106 $\hookrightarrow$ and $\hookleftarrow$ are bijections.
□

Since split preserves both imps and co-imps one would expect that junc does so too. But this is not the case! The proof that split preserves co-imps cannot be transliterated into a proof that junc preserves co-imps (thus emphasising that one has to be very careful with "dualisation" of arguments) and we can only assert that it preserves imps. Nevertheless, $+$ preserves both.

Theorem 107 (Imp and Co-imp Preservation)

(a) $\triangledown$ preserves imps.
(b) If f and g are both co-imps and $f< \sqcap g< \;=\; \perp\!\!\perp$ then $f \triangledown g$ is a co-imp.
(c) $+$ preserves both imps and co-imps.

Proof We leave (a) and (b) as exercises for the reader. (By implication (b) states also that $\triangledown$ does not preserve co-imps in general.) We demonstrate that $+$ preserves co-imps on account of the peculiar contrast in the properties of the two operators.

$$
\begin{array}{cl}
 & R+S \text{ is a co-imp} \\
\equiv & \qquad \{ \text{ definition of co-imp } \} \\
 & I \;\sqsupseteq\; (R+S)^{\cup} \circ (R+S) \\
\equiv & \qquad \{ \;+\text{ is a relator } \} \\
 & I \;\sqsupseteq\; (R^{\cup} \circ R)+(S^{\cup} \circ S) \\
\equiv & \qquad \{ \text{ definition of } + \;\} \\
 & I \;\sqsupseteq\; \hookrightarrow \circ R^{\cup} \circ R \circ \hookrightarrow^{\cup} \;\sqcup\; \hookleftarrow \circ S^{\cup} \circ S \circ \hookleftarrow^{\cup} \\
\Leftarrow & \qquad \{ (59) \} \\
 & I \;\sqsupseteq\; R^{\cup} \circ R \;\sqcup\; S^{\cup} \circ S \\
\equiv & \qquad \{ \text{ plat calculus } \} \\
 & R \text{ is a co-imp } \;\wedge\; S \text{ is a co-imp}
\end{array}
$$

□

8.4 Left and Right Domains

Lemma 105 not only predicts that the injections are co-imps but also that they are total. Formulae for the left domain of the injections are also easy to calculate:

Theorem 108

(a) $\hookrightarrow> \;=\; I$ and $\hookleftarrow> \;=\; I$
(b) $\hookrightarrow< \;=\; I+\perp\!\!\!\perp$ and $\hookleftarrow< \;=\; \perp\!\!\!\perp+I$
□

The next theorem could be said to be the dual to the theorem that the right domains of the projections are equal.

Theorem 109

$$\hookrightarrow< \sqcap \hookleftarrow< \;=\; \perp\!\!\!\perp$$

Proof We show that

$$\hookrightarrow \circ \top\!\!\!\top \sqcap \hookleftarrow \circ \top\!\!\!\top \;=\; \perp\!\!\!\perp$$

The lemma then follows by simple plat calculus.

$$\begin{array}{ll}
 & \neg(\hookleftarrow \circ \top\!\!\!\top) \sqsupseteq \hookrightarrow \circ \top\!\!\!\top \\
\equiv & \qquad \{ \text{ RS rule } \} \\
 & \perp\!\!\!\perp \sqsupseteq \hookrightarrow^{\cup} \circ \hookleftarrow \circ \top\!\!\!\top \\
\equiv & \qquad \{ \text{ theorem 104 } \} \\
 & \textbf{true}
\end{array}$$

□

For many purposes a weaker form of theorem 109 suffices.

Corollary 110

$$\hookrightarrow \sqcap \hookleftarrow \;=\; \perp\!\!\!\perp$$

Proof

$$\begin{array}{ll}
 & \hookrightarrow \sqcap \hookleftarrow \;=\; \perp\!\!\!\perp \\
\equiv & \qquad \{ \text{ lemma 42 } \} \\
 & (\hookrightarrow \sqcap \hookleftarrow)< \;=\; \perp\!\!\!\perp \\
\Leftarrow & \qquad \{ \text{ monotonicity } \} \\
 & \hookrightarrow< \sqcap \hookleftarrow< \;=\; \perp\!\!\!\perp
\end{array}$$

□

In contrast to those for cartesian product the rules for the left and right domains of junc and sum are very simple. Both domain operators distribute over sum, and over junc, but transforming the operator in one case into cup and in the other into sum.

Theorem 111

(a) $(P+Q)> \;=\; P> + Q>$
(b) $(P+Q)< \;=\; P< + Q<$
(c) $(P \triangledown Q)< \;=\; P< \sqcup Q<$
(d) $(P \triangledown Q)> \;=\; P> + Q>$
□

Proof The proofs of (a), (b) and (c) can all be obtained by transliterating the proofs of the corresponding properties of cartesian product. By (14), (d) follows if we can establish that

$$\top\!\!\top \circ (P{>}+Q{>}) \;=\; \top\!\!\top \circ (P \triangledown Q)$$

This we now do.

$$
\begin{array}{ll}
 & \top\!\!\top \circ (P{>}+Q{>}) \\
= & \quad \{ \text{definition of sum, theorem 100} \} \\
 & (\top\!\!\top \circ \hookrightarrow \circ P{>}) \;\triangledown\; (\top\!\!\top \circ \hookleftarrow \circ Q{>}) \\
= & \quad \{ (14) \} \\
 & (\top\!\!\top \circ \hookrightarrow{>} \circ P{>}) \;\triangledown\; (\top\!\!\top \circ \hookleftarrow{>} \circ Q{>}) \\
= & \quad \{ (108) \} \\
 & (\top\!\!\top \circ P{>}) \;\triangledown\; (\top\!\!\top \circ Q{>}) \\
= & \quad \{ (14), \text{theorem 100} \} \\
 & \top\!\!\top \circ (P \triangledown Q)
\end{array}
$$

□

8.5 Unique Extension Property

The counterpart of the *terminality* property of cartesian product is an *initiality* property. Here it is yet stronger: so much so indeed that it warrants a different order of presentation. The key insight is that two components in a junc or sum remain truely disjoint. To be precise:

Theorem 112 (Cancellation Properties)

For all $\unlhd \in \{\sqsubseteq, =, \sqsupseteq\}$,

(a) $\quad P \triangledown Q \unlhd R \triangledown S \;\equiv\; P \unlhd R \;\wedge\; Q \unlhd S$

(b) $\quad P+Q \unlhd R+S \;\equiv\; P \unlhd R \;\wedge\; Q \unlhd S$

Proof

(a)
$$
\begin{array}{ll}
 & P \triangledown Q \unlhd R \triangledown S \\
\Rightarrow & \quad \{ \text{monotonicity} \} \\
 & P \triangledown Q \circ \hookrightarrow \unlhd R \triangledown S \circ \hookrightarrow \;\wedge\; P \triangledown Q \circ \hookleftarrow \unlhd R \triangledown S \circ \hookleftarrow \\
\equiv & \quad \{ \text{computation rules} \} \\
 & P \unlhd R \;\wedge\; Q \unlhd S \\
\Rightarrow & \quad \{ \text{monotonicity} \} \\
 & P \triangledown Q \unlhd R \triangledown S
\end{array}
$$

(b)
$$
\begin{array}{ll}
 & P+Q \unlhd R+S \\
\equiv & \quad \{ \text{definition of sum, (a)} \} \\
 & \hookrightarrow \circ P \unlhd \hookrightarrow \circ R \;\wedge\; \hookleftarrow \circ Q \unlhd \hookleftarrow \circ S \\
\Rightarrow & \quad \{ \text{compose on the left with } \hookrightarrow^{\cup} \text{ and } \hookleftarrow^{\cup}\text{, lemma 105} \} \\
 & P \unlhd R \;\wedge\; Q \unlhd S \\
\Rightarrow & \quad \{ \text{monotonicity} \} \\
 & P+Q \unlhd R+S
\end{array}
$$

□

Corollary 113 (Junc Initiality)

For all $\trianglelefteq \in \{\sqsubseteq, =, \sqsupseteq\}$,

$$P \circ (I{+}I) \trianglelefteq Q \triangledown R \;\equiv\; P \circ \hookrightarrow \trianglelefteq Q \;\wedge\; P \circ \hookleftarrow \trianglelefteq R$$

Proof By the definition of sum, (58) and spec-junc fusion, (100),

$$P \circ (I{+}I) \;=\; (P \circ \hookrightarrow) \triangledown (P \circ \hookleftarrow)$$

Initiality thus follows immediately.
$\square$

8.6 Naturality Properties

The naturality properties of the two injections are stronger than those of the projections.

Theorem 114 (Naturality of Left and Right Injection)
For all specs R and S,

(a) $\quad \hookrightarrow \;\in\; R{+}S \overset{\cdot}{<\!\sim\!>} R$
(b) $\quad \hookleftarrow \;\in\; R{+}S \overset{\cdot}{<\!\sim\!>} S$

Proof Immediate from the computation rules and the definition of $\overset{\cdot}{<\!\sim\!>}$.
$\square$

The naturality property of junc is also stronger.

Theorem 115 (Naturality of Junc and Sum)
For all specs R, S, T and U and all $\overset{\cdot}{\sim} \;\in\; \{\overset{\cdot}{<\!\sim}, \overset{\cdot}{\sim\!>}, \overset{\cdot}{<\!\sim\!>}\}$,

(a) $\quad \triangledown \;\in\; (R \overset{\cdot}{\sim} S{+}T) \overset{\cdot}{<\!\sim} (R \overset{\cdot}{\sim} S) \times (R \overset{\cdot}{\sim} T)$
(b) $\quad + \;\in\; (R{+}S \overset{\cdot}{\sim} T{+}U) \overset{\cdot}{<\!\sim} (R \overset{\cdot}{\sim} T) \times (S \overset{\cdot}{\sim} U)$

Proof In the following proof we use $\trianglelefteq$ to stand for $\sqsupseteq$, $\sqsubseteq$ or $=$ depending on the value of $\overset{\cdot}{\sim}$. (*Cf* the definitions of the three naturality operators.)

$$\begin{array}{ll}
 & \triangledown \;\in\; (R \overset{\cdot}{\sim} S{+}T) \overset{\cdot}{<\!\sim} (R \overset{\cdot}{\sim} S) \times (R \overset{\cdot}{\sim} T) \\
\equiv & \qquad \{ \text{ theorem 38 } \} \\
 & \forall (U,V,W,X :: \\
 & \quad U \triangledown V \,\langle R \overset{\cdot}{\sim} S{+}T \rangle\, W \triangledown X \;\;\Leftarrow\;\; U \,\langle R \overset{\cdot}{\sim} S \rangle\, W \;\wedge\; V \,\langle R \overset{\cdot}{\sim} T \rangle\, X \\
 &)
\end{array}$$

We now continue with the quantified expression.

$$\begin{array}{ll}
 & U \triangledown V \,\langle R \overset{\cdot}{\sim} S{+}T \rangle\, W \triangledown X \;\;\Leftarrow\;\; U \,\langle R \overset{\cdot}{\sim} S \rangle\, W \;\wedge\; V \,\langle R \overset{\cdot}{\sim} T \rangle\, X \\
\equiv & \qquad \{ \text{ definition of } \overset{\cdot}{\sim} \} \\
 & R \circ (W \triangledown X) \trianglelefteq (U \triangledown V) \circ (S{+}T) \\
 & \quad \Leftarrow \quad R \circ W \trianglelefteq U \circ S \;\;\wedge\;\; R \circ X \trianglelefteq V \circ T \\
\equiv & \qquad \{ \text{ theorems 100 and 98 } \} \\
 & (R \circ W) \triangledown (R \circ X) \trianglelefteq (U \circ S) \triangledown (V \circ T) \\
 & \quad \Leftarrow \quad R \circ W \trianglelefteq U \circ S \;\;\wedge\;\; R \circ X \trianglelefteq V \circ T \\
\equiv & \qquad \{ \text{ monotonicity } \}
\end{array}$$

true

The verification of (b) proceeds in the same way.

□

8.7 Junctivity Properties

The finite junctivity properties of disjoint sum are stronger than those for cartesian product:

Theorem 116 (Junc/Sum-Cup/Cap Abide Laws)

(a) $(P \triangledown Q) \sqcup (R \triangledown S) = (P \sqcup R) \triangledown (Q \sqcup S)$

(b) $(P+Q) \sqcup (R+S) = (P \sqcup R)+(Q \sqcup S)$

(c) $(P \triangledown Q) \sqcap (R \triangledown S) = (P \sqcap R) \triangledown (Q \sqcap S)$

(d) $(P+Q) \sqcap (R+S) = (P \sqcap R)+(Q \sqcap S)$

Proof Since (d) is a slight variation on (c), while (a) and (b) are trivial, we only prove (c):

$$
\begin{array}{ll}
 & (P \triangledown Q) \;\sqcap\; (R \triangledown S) \\
= & \quad \{\, (58) \,\} \\
 & (P \circ \hookrightarrow_{\cup} \sqcup\; Q \circ \hookleftarrow_{\cup}) \;\sqcap\; (R \circ \hookrightarrow_{\cup} \sqcup\; S \circ \hookleftarrow_{\cup}) \\
= & \quad \{\, \text{distribution} \,\} \\
 & (P \circ \hookrightarrow_{\cup} \sqcap\; R \circ \hookrightarrow_{\cup}) \;\sqcup\; (Q \circ \hookleftarrow_{\cup} \sqcap\; S \circ \hookleftarrow_{\cup}) \;\sqcup \\
 & (Q \circ \hookleftarrow_{\cup} \sqcap\; R \circ \hookrightarrow_{\cup}) \;\sqcup\; (P \circ \hookrightarrow_{\cup} \sqcap\; S \circ \hookleftarrow_{\cup}) \\
= & \quad \{\, (109)\,,\; \hookrightarrow_{\cup} \text{ and } \hookleftarrow_{\cup} \text{ are imps} \,\} \\
 & (P \sqcap R) \circ \hookrightarrow_{\cup} \;\sqcup\; (Q \sqcap S) \circ \hookleftarrow_{\cup} \\
= & \quad \{\, (58) \,\} \\
 & (P \sqcap R) \triangledown (Q \sqcap S)
\end{array}
$$

□

Again, more can be shown: The junction and the sum are positively cap-junctive and universally cup-junctive. Hence they are cap- and cup-continuous.

Theorem 117 Let $\mathcal{V}$ be a bag of pairs (V_λ, V_ρ) of specs, $L = \sqcup(V : V \in \mathcal{V} : V_\lambda)$ and $R = \sqcup(V : V \in \mathcal{V} : V_\rho)$. Then

(a) $L \triangledown R \;=\; \sqcup(V : V \in \mathcal{V} : V_\lambda \triangledown V_\rho)$

(b) $L+R \;=\; \sqcup(V : V \in \mathcal{V} : V_\lambda + V_\rho)$

□

Theorem 118 Let $\mathcal{V}$ be a non-empty bag of pairs (V_λ, V_ρ) of specs, $L = \sqcap(V : V \in \mathcal{V} : V_\lambda)$ and $R = \sqcap(V : V \in \mathcal{V} : V_\rho)$. Then

(a) $L \triangledown R \;=\; \sqcap(V : V \in \mathcal{V} : V_\lambda \triangledown V_\rho)$

(b) $L+R \;=\; \sqcap(V : V \in \mathcal{V} : V_\lambda + V_\rho)$

Proof

$$
\begin{array}{ll}
 & L \triangledown R \\
\sqsubseteq & \quad \{\, \triangledown \text{ is monotonic} \,\} \\
 & \sqcap(V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow_{\cup} \sqcup\; V_\rho \circ \hookleftarrow_{\cup}) \\
\sqsubseteq & \quad \{\, \text{quantifier calculus} \,\}
\end{array}
$$

$$
\begin{array}{ll}
 & \sqcap(V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow^{\cup}) \sqcup \sqcap(V : V \in \mathcal{V} : V_\rho \circ \hookleftarrow^{\cup}) \\
 & \quad \sqcup (\top\!\!\top \circ \hookrightarrow^{\cup} \sqcap \top\!\!\top \circ \hookleftarrow^{\cup}) \\
= & \quad \{ (109) \} \\
 & \sqcap(V : V \in \mathcal{V} : V_\lambda \circ \hookrightarrow^{\cup}) \sqcup \sqcap(V : V \in \mathcal{V} : V_\rho \circ \hookleftarrow^{\cup}) \\
= & \quad \{ \hookrightarrow^{\cup} \text{ is an imp and } \mathcal{V} \text{ is non-empty} \} \\
 & \sqcap(V : V \in \mathcal{V} : V_\lambda) \circ \hookrightarrow^{\cup} \sqcup \sqcap(V : V \in \mathcal{V} : V_\rho) \circ \hookleftarrow^{\cup} \\
= & \quad \{ (58) \} \\
 & L \triangledown R
\end{array}
$$

□

9 Natural Simulations and Natural Isomorphisms

To summarise, we now have one non-trivial monotype and two binary relators. Unary relators can be derived from these by fixing one of the arguments to a monotype; ternary relators, quaternary relators etc. can be obtained by composing them in appropriate ways; and new monotypes can be obtained by applying existing relators to existing monotypes. For example, $\mathbf{1}+\mathbf{1}$ and $\mathbf{1} \times (\mathbf{1}+\mathbf{1})$ are monotypes, and the functions $\mathbf{1}+$ and $(\mathbf{1} \times \mathbf{1})+$ are unary relators. Relators and monotypes built in this way are called *polynomial*. This, however, is just the foundation. It is only now that our theory can really begin.

In this section we make a modest start to showing the ease with which certain calculations can be made within the theory. At the same time we formulate a number (at this point in time 2!) of concepts of particular relevance to data reification.

Definition 119 (Natural Simulation) Relator F is said to (naturally) simulate relator G if and only if there exists a spec γ such that

(a) for all specs R, $\gamma \in F.R \mathrel{<\!\!\cdot\!\!>} G.R$
(b) γ is an imp
(c) $\gamma> \;=\; G.I$

The spec γ itself is called a natural simulation.
□

Note that the combination of conditions (a) and (b) implies that $\gamma< \;\sqsubseteq\; F.I$.

Definition 120 (Natural Isomorphism) Relators F and G are said to be naturally isomorphic if and only if there exists a spec γ such that

(a) for all specs R, $\gamma \in F.R \mathrel{<\!\!\cdot\!\!>} G.R$
(b) γ is a bijection
(c) $\gamma< \;=\; F.I$ and $\gamma> \;=\; G.I$

The spec γ itself is called a natural isomorphism.
□

Remark We have to admit that, at this stage in our research, we are not sure whether it is desirable to weaken condition (a) by replacing $\mathrel{<\!\!\cdot\!\!>}$ by $\mathrel{<\!\!\cdot}$. We stick to the above definition at this point in time because we do not know any examples of natural simulations according to this weaker definition that are not also natural simulations according to the stronger definition.

A simulation satisfying the weaker definition is referred to below as an *up-simulation. End of Remark*

Examples of natural isomorphisms are provided by the two injections $\hookrightarrow$ and $\hookleftarrow$. The former is a natural isomorphism between the relator $(+\perp\!\!\perp)$ and the identity relator. I.e.

$$\begin{array}{l} \hookrightarrow\ \in\ R+\perp\!\!\perp \leftrightarrow R, \text{ for all specs } R \\ \hookrightarrow\ \text{ is a bijection, and} \\ \hookrightarrow< \;=\; I+\perp\!\!\perp \quad\text{and}\quad \hookrightarrow> \;=\; I \end{array}$$

Similarly, the latter is an isomorphism between the the relator $(\perp\!\!\perp+)$ and the identity relator. The injections are also examples of natural simulations: $\hookrightarrow$ is, for example, a natural simulation of the identity relator by the relator $+\mathbf{1}$. (In general any monotype may be used in place of $\mathbf{1}$.)

As might be expected, both natural simulations and natural isomorphisms enjoy many simple but powerful algebraic properties. In later versions of this paper it is our intention to document some of them. For the time being, however, we leave the reader the pleasure of their discovery. Let us proceed to more significant examples. We begin with the most complicated, basic example of a natural isomorphism.

Consider the ternary relators defined by

$$\begin{array}{lcl} R,S,T & \mapsto & R\times(S+T) \\ R,S,T & \mapsto & (R\times S)+(R\times T) \end{array}$$

Our objective is to show that the two relators are naturally isomorphic.

To complete this task we must exhibit a spec, γ, satisfying three quite strong conditions. We can make progress in this task by temporarily setting aside two of the conditions, *constructing* γ to satisfy the remaining condition, and then (hopefully) *verifying* that it satisfies the two other conditions. The condition singled out should be the one that leaves the least freedom to manoeuvre, in this case clearly condition (a). What we shall now demonstrate is how systematically this can be done using the rules we have given for the naturally polymorphic type of the operators we have introduced.

Here then is the construction of the desired natural isomorphism. Assume R, S, and T are arbitrary specs. Then

by construction of γ:

$$\begin{array}{ll} & \gamma\ \in\ R\times(S+T) \leftrightarrow (R\times S)+(R\times T) \\ \Leftarrow & \quad \{\ \text{naturality of } \triangledown,\ \gamma := \gamma_1 \triangledown \gamma_2\ \} \\ & \gamma_1\ \in\ R\times(S+T) \leftrightarrow R\times S \\ \wedge & \gamma_2\ \in\ R\times(S+T) \leftrightarrow R\times T \\ \Leftarrow & \quad \{\ \text{naturality of product, } I\ \in\ R \leftrightarrow R, \\ & \qquad \gamma_1 := I\times\gamma_1,\ \ \gamma_2 := I\times\gamma_2\ \} \\ & \gamma_1\ \in\ S+T \leftrightarrow S \quad\wedge\quad \gamma_2\ \in\ S+T \leftrightarrow T \\ \Leftarrow & \quad \{\ \text{naturality of the injections}\ \} \\ & \gamma_1 = \hookrightarrow \quad\wedge\quad \gamma_2 = \hookleftarrow \end{array}$$

Thus the constructed spec is γ where

$$\gamma \;=\; (I\times\hookrightarrow)\triangledown(I\times\hookleftarrow)$$

It remains to show that γ is a bijection and has the correct left and right domains. The verifications are straightforward, but we give them nonetheless as proof of the pudding.

First, we assert that γ is a bijection. That it is an imp follows because it is built out of imps using imp-preserving operators. Since junc is not necessarily co-imp preserving we need to take further steps to show that it is a co-imp.

$$\begin{array}{ll}
 & \gamma \text{ is a co-imp} \\
\Leftarrow & \quad \{ \text{ theorem 107(b) } \} \\
 & (I \times \hookrightarrow \text{ and } I \times \hookleftarrow \text{ are co-imps }) \\
 & \wedge\ (I \times \hookrightarrow)< \sqcap (I \times \hookleftarrow)< \ =\ \perp\!\!\perp \\
\equiv & \quad \{ \text{ theorem } 80 \ \} \\
 & (I \times \hookrightarrow)< \sqcap (I \times \hookleftarrow)< \ =\ \perp\!\!\perp \\
\equiv & \quad \{ \text{ theorem } 84 \ \} \\
 & (I \times \hookrightarrow<) \sqcap (I \times \hookleftarrow<) \ =\ \perp\!\!\perp \\
\equiv & \quad \{ \text{ lemma 94(b) } \} \\
 & I \times (\hookrightarrow< \sqcap \hookleftarrow<) \ =\ \perp\!\!\perp \\
\equiv & \quad \{ \text{ theorem 109, product is strict } \} \\
 & \textbf{true}
\end{array}$$

We now calculate the left domain of γ.

$$\begin{array}{ll}
 & \gamma< \\
= & \quad \{ \text{ definition of } \gamma, \text{ theorem 111(c) } \} \\
 & (I \times \hookrightarrow)< \ \sqcup\ (I \times \hookleftarrow)< \\
= & \quad \{ \text{ theorems 84 and 108 } \} \\
 & I \times (I + \perp\!\!\perp) \ \sqcup\ I \times (\perp\!\!\perp + I) \\
= & \quad \{ \text{ lemma 95(e) } \} \\
 & I \times ((I + \perp\!\!\perp) \sqcup (\perp\!\!\perp + I)) \\
= & \quad \{ \text{ definition of } + \ \} \\
 & I \times (I + I)
\end{array}$$

Finally, we calculate the right domain of γ.

$$\begin{array}{ll}
 & \gamma> \\
= & \quad \{ \text{ definition of } \gamma, \text{ theorem 111(d) } \} \\
 & (I \times \hookrightarrow)> \ +\ (I \times \hookleftarrow)> \\
= & \quad \{ \text{ theorems 84 and 108 } \} \\
 & (I \times I) + (I \times I)
\end{array}$$

This completes the verification.

The point of discussing this example in so much detail is to emphasise the importance of type considerations in *constructing* specs having prescribed properties. (This is a somewhat different emphasis than that which one encounters most frequently. Wadler [17], for example, discusses the use of natural polymorphism to *infer* properties of already constructed functions.) There is, however, yet more that can be said about the bijection γ that we have constructed that so far as we know is not predicted by any naturality theorem and yet seems "obvious" from type considerations. The properties that we allude to record its behaviour with respect to the two morphisms split and junc. Before stating and proving the properties we need to interpose a truly remarkable and elegant law permitting an exchange of split for junc and vice-versa.

Theorem 121 (Split-Junc Abide Law)

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

Proof We aim to use the initiality property (theorem 113) of junc. First note that

$$
\begin{array}{ll}
 & (R \triangledown S) \vartriangle (T \triangledown U) \circ I{+}I \\
= & \quad \{ \ I{+}I \text{ is a monotype and thus an imp, } (70) \ \} \\
 & (R \triangledown S \circ I{+}I) \vartriangle (T \triangledown U \circ I{+}I) \\
= & \quad \{ \ \text{split fusion } (68) \ \} \\
 & (R \triangledown S) \vartriangle (T \triangledown U)
\end{array}
$$

Hence:

$$
\begin{array}{ll}
 & (R \triangledown S) \vartriangle (T \triangledown U) \ = \ (R \vartriangle T) \triangledown (S \vartriangle U) \\
\equiv & \quad \{ \ \text{theorem 113 combined with the above} \ \} \\
 & (R \triangledown S) \vartriangle (T \triangledown U) \circ \hookrightarrow \ = \ R \vartriangle T \\
 & \wedge \quad (R \triangledown S) \vartriangle (T \triangledown U) \circ \hookleftarrow \ = \ S \vartriangle U
\end{array}
$$

Continuing now with just one of the conjuncts in the last expression we calculate:

$$
\begin{array}{ll}
 & (R \triangledown S) \vartriangle (T \triangledown U) \circ \hookrightarrow \\
= & \quad \{ \ \hookrightarrow \text{ is an imp, split-imp fusion} \ \} \\
 & (R \triangledown S \circ \hookrightarrow) \ \vartriangle \ (T \triangledown U \circ \hookrightarrow) \\
= & \quad \{ \ \text{junc-computation} \ \} \\
 & R \ \vartriangle \ T
\end{array}
$$

The other conjunct being dealt with in a similar way our proof is now complete.
□

The properties of the natural isomorphism γ that we anticipated above can now be given.

Theorem 122

(a) $\gamma \circ (R \vartriangle S){+}(R \vartriangle T) \ = \ (R \circ I \triangledown I) \vartriangle (S{+}T)$

(b) $R \times (S \triangledown T) \circ \gamma \ = \ (R \times S) \ \triangledown \ (R \times T)$

Proof

$$
\begin{array}{lll}
\text{(a)} & & \gamma \circ (R \vartriangle S){+}(R \vartriangle T) \\
 & = & \quad \{ \ \text{definition of } \gamma \text{, junc fusion theorem 98(a)} \ \} \\
 & & (I \times \hookrightarrow \circ \ R \vartriangle S) \ \triangledown \ (I \times \hookleftarrow \circ \ R \vartriangle T) \\
 & = & \quad \{ \ \text{split fusion theorem 68(a)} \ \} \\
 & & (R \vartriangle (\hookrightarrow \circ S)) \ \triangledown \ (R \vartriangle (\hookleftarrow \circ T)) \\
 & = & \quad \{ \ \text{abides law } (121) \ \} \\
 & & (R \triangledown R) \ \vartriangle \ ((\hookrightarrow \circ S) \ \triangledown \ (\hookleftarrow \circ T)) \\
 & = & \quad \{ \ \text{junc fusion (98a), definition of sum } (58) \ \} \\
 & & (R \circ I \triangledown I) \ \vartriangle \ (S{+}T)
\end{array}
$$

$$
\begin{array}{lll}
\text{(b)} & & R \times (S \triangledown T) \circ \gamma \\
 & = & \quad \{ \ \text{definition of } \gamma \text{, spec-junc fusion } (100) \ \} \\
 & & (R \times (S \triangledown T) \circ I \times \hookrightarrow) \ \triangledown \ (R \times (S \triangledown T) \circ I \times \hookleftarrow)) \\
 & = & \quad \{ \ \times \text{ is a relator, junc computation rules } (103) \ \} \\
 & & (R \times S) \ \triangledown \ (R \times T)
\end{array}
$$

□

Natural isomorphisms seem to receive scant attention in the category theory literature, often being relegated to a brief exercise. This is somewhat unfortunate because it deemphasises their importance and it means that no guidance is given on how to construct them. We also relegate the construction of several basic natural isomorphisms to the present set of exercises, not because they are unimportant but because by doing them the reader may be enabled to make a judgement on the effectiveness of the calculus developed thus far.

It is useful to begin by listing the elementary natural isomorphisms. For this purpose we use a home-grown, but hopefully self-evident, lambda notation.

(123) $\lambda(R :: \perp\!\!\perp) \cong \lambda(R :: R \times \perp\!\!\perp)$

(124) $\lambda(R :: R + \perp\!\!\perp) \cong \lambda(R :: R)$

(125) $\lambda(R,S :: R+S) \cong \lambda(R,S :: S+R)$

(126) $\lambda(R,S,T :: R+(S+T)) \cong \lambda(R,S,T :: (R+S)+T)$

(127) $\lambda(R,S :: R \times S) \cong \lambda(R,S :: S \times R)$

(128) $\lambda(R,S,T :: R \times (S \times T)) \cong \lambda(R,S,T :: (R \times S) \times T)$

(129) $\lambda(R :: R) \cong \lambda(R :: R \times \mathbf{1})$

(130) $\lambda(R,S,T :: R \times (S+T)) \cong \lambda(R,S,T :: (R \times S)+(R \times T))$

Of these (123) is trivial (the isomorphism is $\perp\!\!\perp$ itself) and (124) and (130) we have already discussed. Hints on how to prove (125)-(129) are given below. Of course the reader may wish to ignore the hints altogether.

Hints: Isomorphisms (125) and (126) can be constructed using the same strategy as that used to construct (130). In the case of (125) the very short calculations that are necessary can be made yet shorter by noting that the constructed isomorphism is its own reverse.

Isomorphisms (127) and (128) require a somewhat different strategy. The reason is that the natural isomorphism properties of split and product only help in the construction of up-simulations (elements of $R \leftsquigarrow S$ for given R and S) and not natural transformations. Moreover, whereas the calculation of the right domain of a split is straightforward, the calculation of its left domain is not (compare 85(a) with 85(b)). To add to this, split preserves imps but does not preserve co-imps. One may avoid all these difficulties by constructing two up-simulations, one of the left-side relator by the right-side relator and one of the right-side relator by the left-side relator. (In the case of (127) these "two" up-simulations obviously coincide.) For this purpose the naturality properties are used. One then proves that the first is the reverse of the second. It then suffices to prove that both are imps and to calculate the right domains of each.

The proof of (129) is a case apart. Note that $\ll$ is an up-simulation of the identity relator by $\times\mathbf{1}$. Try restricting $\ll$ so that its right domain is $I \times \mathbf{1}$ and then verify that your conjectured isomorphism meets all the requirements. *End of hints*

Having completed this task you should be able to verify the following properties of the constructed isomorphisms. (The names $\alpha_1 \ldots \alpha_8$ have been given to the isomorphisms in order of their appearance in the list above.)

(131) $\perp\!\!\perp = \alpha_1 \circ R \vartriangle \perp\!\!\perp$

(132) $R \triangledown \perp\!\!\perp \circ \alpha_2 = R$

(133) $R \triangledown S \circ \alpha_3 = S \triangledown R$

(134) $R \triangledown (S \triangledown T) \circ \alpha_4 = (R \triangledown S) \triangledown T$

(135) $R \vartriangle S = \alpha_5 \circ S \vartriangle R$

$$\begin{array}{lrcl} (136) & R \vartriangle (S \vartriangle T) & = & \alpha_6 \circ (R \vartriangle S) \vartriangle T \\ (137) & R & = & \alpha_7 \circ R \vartriangle ! \end{array}$$

Rest assured: all have very trivial proofs. Note the pattern: the relators $+$ and $\times$ have been systematically replaced by their corresponding morphism and $\mathbf{1}$ has been replaced by its morphism !. The α's are eaten up on the right side by a junc and on the left side by a split.

Consider now the quaternary relators F and G, respectively, defined by

$$R,S,T,U \quad \mapsto \quad (R{+}S) \times (T{+}U)$$

$$R,S,T,U \quad \mapsto \quad (R \times T) + (S \times U)$$

We conclude this section by showing that F simulates G.

We begin by constructing γ satisfying requirement (a) of a natural simulation (see definition 119).

$$\begin{array}{ll} & \gamma \;\in\; (R{+}S) \times (T{+}U) \;\Leftrightarrow\; (R \times T) + (S \times U) \\ \Leftarrow & \quad \{\ \gamma := \beta_1 \triangledown \beta_2 \text{ , naturality of junc }\} \\ & \beta_1 \;\in\; (R{+}S) \times (T{+}U) \;\Leftrightarrow\; R \times T \\ & \wedge\;\; \beta_2 \;\in\; (R{+}S) \times (T{+}U) \;\Leftrightarrow\; S \times U \\ \Leftarrow & \quad \{\ \beta_1 := \alpha_1 \times \alpha_2 \text{ , } \beta_2 := \alpha_3 \times \alpha_4 \text{ , naturality of product }\} \\ & \alpha_1 \;\in\; R{+}S \;\Leftrightarrow\; R \;\wedge\; \alpha_2 \;\in\; T{+}U \;\Leftrightarrow\; T \\ & \wedge\;\; \alpha_3 \;\in\; R{+}S \;\Leftrightarrow\; S \;\wedge\; \alpha_4 \;\in\; T{+}U \;\Leftrightarrow\; U \\ \Leftarrow & \quad \{\ \text{naturality of the injections }\} \\ & \alpha_1 \;=\; \alpha_2 \;=\; \hookrightarrow \;\;\wedge\;\; \alpha_3 \;=\; \alpha_4 \;=\; \hookleftarrow \end{array}$$

We have thus shown that

$$(\hookrightarrow \times \hookrightarrow) \triangledown (\hookleftarrow \times \hookleftarrow) \;\in\; (R{+}S) \times (T{+}U) \;\Leftrightarrow\; (R \times T){+}(S \times U)$$

We continue to call the constructed spec γ.

Clearly, γ is an imp; it is, however, not a co-imp and cannot therefore be an isomorphism. Calculation of its right domain using (111), (84) and (108) is straightforward. We obtain

$$\gamma{>} \;=\; (I \times I) + (I \times I)$$

as required. The verification that F simulates G is thus complete.

In just the same way that we explored the behaviour of natural isomorphisms with respect to split and junc, it is useful to explore further the properties of the simulation γ. First, in a matter of a few steps using junc-sum and product-split fusion followed by the junc-split abide law and the definition of sum, one obtains

$$\gamma \circ (R \vartriangle T){+}(S \vartriangle U) \;=\; (R{+}S) \vartriangle (T{+}U)$$

Second, using the definition of product, the junc-split abide law and the definition of sum one obtains:

$$\gamma \;=\; (\ll{+}\ll) \vartriangle (\gg{+}\gg)$$

which is a better form of γ for the final calculation which is to verify that

$$(R \triangledown S) \times (T \triangledown U) \circ \gamma \quad = \quad (R \times T) \triangledown (S \times U)$$

(This calculation also takes only a few steps and involves using the fusion laws and the definition of product.)

This concludes our discussion of natural simulations and of the elementary properties of the polynomial relators.

References

[1] R.C. Backhouse. On a relation on functions. In W.H.J. Feijen, A.J.M. van Gasteren, D. Gries, and J. Misra, editors, *Beauty is our Business*. Springer-Verlag, 1990.

[2] R.C. Backhouse, Bruin P. de, P. Hoogendijk, G. Malcolm, Voermans T.S., and J. van der Woude. A relational theory of datatypes. In preparation: copies of draft available on request, 1991.

[3] R.C. Backhouse, Bruin P. de, G. Malcolm, Voermans T.S., and J. van der Woude. Relational catamorphisms. In Möller B., editor, *Proceedings of the IFIP TC2/WG2.1 Working Conference on Constructing Programs*. Elsevier Science Publishers B.V., 1991.

[4] R. Berghammer and H. Zierer. Relational algebraic semantics of deterministic and nondeterministic programs. *Theoretical Computer Science*, 43:123–147, 1986.

[5] Garrett Birkhoff. *Lattice Theory*, volume 25 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, Rhode Island, 3rd edition, 1948.

[6] P.J. de Bruin. Naturalness of polymorphism. Technical Report CS8916, Department of Mathematics and Computing Science, University of Groningen, 1989.

[7] E.W. Dijkstra and W.H.J. Feijen. *Een Methode van Programmeren*. Academic Service, Den Haag, 1984. Also available as *A Method of Programming*, Addison-Wesley, Reading, Mass., 1988.

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

[9] C.A.R. Hoare. A couple of novelties in the propositional calculus. *Zeitschr. fuer Math. Logik und Grundlagen der Math.*, 31(2):173–178, 1985.

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

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

[12] Z. Manna and R. Waldinger. *The Logical Basis for Computer Programming, Volume 1: Deductive Reasoning*. Addison-Wesley, 1985.

[13] L. Meertens. Constructing a calculus of programs. In J.L.A. van de Snepscheut, editor, *Conference on the Mathematics of Program Construction*, pages 66–90. Springer-Verlag LNCS 375, 1989.

[14] J.C. Reynolds. Types, abstraction and parametric polymorphism. In R.E. Mason, editor, *IFIP '83*, pages 513–523. Elsevier Science Publishers, 1983.

[15] Willem Paul de Roever Jr. *Recursive program schemes: semantics and proof theory*. PhD thesis, Free University, Amsterdam, January 1974.

[16] G. Schmidt and T. Ströhlein. Relation algebras: Concept of points and representability. *Discrete Mathematics*, 54:83–92, 1985.

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

90/1	W.P.de Roever- H.Barringer- C.Courcoubetis-D.Gabbay R.Gerth-B.Jonsson-A.Pnueli M.Reed-J.Sifakis-J.Vytopil P.Wolper	Formal methods and tools for the development of distributed and real time systems, p. 17.
90/2	K.M. van Hee P.M.P. Rambags	Dynamic process creation in high-level Petri nets, pp. 19.
90/3	R. Gerth	Foundations of Compositional Program Refinement - safety properties - , p. 38.
90/4	A. Peeters	Decomposition of delay-insensitive circuits, p. 25.
90/5	J.A. Brzozowski J.C. Ebergen	On the delay-sensitivity of gate networks, p. 23.
90/6	A.J.J.M. Marcelis	Typed inference systems : a reference document, p. 17.
90/7	A.J.J.M. Marcelis	A logic for one-pass, one-attributed grammars, p. 14.
90/8	M.B. Josephs	Receptive Process Theory, p. 16.
90/9	A.T.M. Aerts P.M.E. De Bra K.M. van Hee	Combining the functional and the relational model, p. 15.
90/10	M.J. van Diepen K.M. van Hee	A formal semantics for Z and the link between Z and the relational algebra, p. 30. (Revised version of CSNotes 89/17).
90/11	P. America F.S. de Boer	A proof system for process creation, p. 84.
90/12	P.America F.S. de Boer	A proof theory for a sequential version of POOL, p. 110.
90/13	K.R. Apt F.S. de Boer E.R. Olderog	Proving termination of Parallel Programs, p. 7.
90/14	F.S. de Boer	A proof system for the language POOL, p. 70.
90/15	F.S. de Boer	Compositionality in the temporal logic of concurrent systems, p. 17.
90/16	F.S. de Boer C. Palamidessi	A fully abstract model for concurrent logic languages, p. p. 23.
90/17	F.S. de Boer C. Palamidessi	On the asynchronous nature of communication in logic logic languages: a fully abstract model based on sequences, p. 29.